Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Matemáticas y Física

Maestría en Ciencias de Datos

Aplicación de modelos de Machine Learning para Nowcasting: una solución para la planeación en la cadena de suministro en la industria electrónica

---

Tesis para obtener el **GRADO** de
MAESTRÍA EN CIENCIAS DE DATOS

Tesis presentada por:
Erika Ramírez Ortega

Director de tesis:
Dr. Luis Raul Rodríguez Reyes

Tlaquepaque, Jalisco, Mayo, 2026

Instituto Tecnológico y de Estudios Superiores de Occidente

Reconocimiento de validez oficial de estudios de nivel superior según acuerdo secretarial 15018, publicado en el Diario Oficial de la Federación del 29 de noviembre de 1976.

Departamento de Matemáticas y Física

Formulario de Aprobación de la Maestría en Ciencias de Datos

Título de la Tesis: **Aplicación de modelos de Machine Learning para Nowcasting: una solución para la planeación en la cadena de suministro en la industria electrónica**

Autor: **Erika Ramírez Ortega**

Tesis aprobada para completar todos los requisitos de grado de la Maestría en Ciencias de Datos.

Director de Tesis, **Dr. Luis Raul Rodríguez Reyes**

Revisor de Tesis, **Angel Samaniego Alcantar**

Revisor de Tesis, **María Del Rosario Ruíz Hernandez**

Tlaquepaque, Jalisco, Mayo, 2026

Aplicación de modelos de Machine Learning para Nowcasting: una solución para la planeación en la cadena de suministro en la industria electrónica

Erika Ramírez Ortega

Abstract

This research focuses on forecasting the **Producer Price Index (PPI)** of **integrated circuits**. The study proposes a *nowcasting* approach to estimate the value of this monthly frequency variable by integrating **daily frequency** independent variables, allowing for the capture of market dynamics in real-time.

In contrast to traditional approaches, this thesis moves away from conventional econometric models to prioritize the use of **Machine Learning** architectures, such as: **Support Vector Machines (SVM), Decision Trees, Bagging, and Random Forests**; additionally integrating the **Long Short-Term Memory (LSTM)** neural network due to its ability to model complex sequential dependencies.

To guarantee the reliability of the predictions and determine the technical feasibility of their implementation, a rigorous validation framework was applied. This includes **Bootstrap Reality Check** to compare performance against baseline models, the calculation of the **Probability of Backtest Overfitting (PBO)** to ensure that the predictive capacity of the selected models is statistically sound and not the result of spurious fitting to historical noise, and finally, the **Rolling Walk-Forward** technique, which simulates the model's performance in a real production environment through sliding windows.

Keywords: Nowcasting, Machine Learning, Producer Price Index (PPI), Integrated Circuits, Supply Chain.

Aplicación de modelos de Machine Learning para Nowcasting: una solución para la planeación en la cadena de suministro en la industria electrónica

Erika Ramírez Ortega

Resumen

Este trabajo de investigación se centra en el pronóstico del **Índice de Precios del Productor (IPP)** de los **circuitos integrados**. El estudio propone un enfoque de *nowcasting* para estimar el valor de esta variable de frecuencia mensual mediante la integración de variables independientes de **frecuencia diaria**, permitiendo capturar la dinámica del mercado en tiempo real.

A diferencia de las aproximaciones tradicionales, esta tesis se aleja de los modelos econométricos convencionales para priorizar el uso de arquitecturas de **Machine Learning**, como lo son: **Maquinas de Soporte Vectorial (SVM), Árboles de Decisión, Bolsa de Árboles y Bosque Aleatorios**; integrando además la red neuronal de **Memoria a Largo y Corto Plazo (LSTM)** por su capacidad para modelar dependencias secuenciales complejas.

Para garantizar la fiabilidad de las predicciones y determinar la conveniencia técnica de su implementación, se aplicó un marco de validación riguroso. Este incluye la prueba de **Bootstrap Reality Check** para comparar el rendimiento frente a modelos base, el cálculo de la **Probabilidad de Overfitting del Backtest (PBO)**, asegurando que la capacidad predictiva de los modelos seleccionados sea estadísticamente sólida y no producto de un ajuste espurio al ruido histórico y por último, la técnica de **Rolling Walking-Forward**, la cual simula el desempeño del modelo en un entorno de producción real mediante ventanas deslizantes.

Palabras clave: Nowcasting, Aprendizaje Automático, Índice de Precios del Productor (IPP), Circuitos Integrados, Cadena de Suministro.

Índice

Índice de figuras

Página

Índice de cuadros

Página

Dedicated to:

I dedicate this work, first and foremost, to God, for granting me the wisdom and strength necessary to embark on and complete this academic journey.

To my parents, Francisca and Guillermo, to my sister, Nancy, and to her husband, Víctor. Thank you for listening to me with such patience, even when this world felt so foreign to you; your words of encouragement were the fuel that kept me from giving up.

To Flor, for being the first person to support me in this decision, for being my confidant and partner-in-crime through every challenge I faced. To Katya and Andrea, for understanding my absences and celebrating my small victories as if they were your own; thank you for being exemplary women and role models who inspire me every day. To Laila, Paola, and Ana, who despite the distance, were present throughout my learning process and always had the right words of encouragement to motivate me.

To all my classmates, for making this academic adventure a shared experience full of unforgettable moments.

To the institution, for providing the necessary support and for placing in my path professors who added knowledge and value to my professional development day after day.

Finally, to the Erika of 2020: today I can look back with pride and tell myself with absolute certainty: “It was all worth it.”

Dedicado a:

Dedico este trabajo, primeramente, a Dios, por brindarme la sabiduría y la fortaleza necesaria para emprender y culminar este camino académico.

A mis padres, Francisca y Guillermo, a mi hermana, Nancy, y a su esposo, Víctor. Gracias por escucharme con paciencia a pesar de lo ajeno que les resultaba este mundo; sus palabras de aliento fueron el combustible que me impidió rendirme.

A Flor, por ser la primera persona en apoyarme en esta decisión, por ser mi confidente y cómplice en cada uno de los retos que enfrenté. A Katya y Andrea, por comprender mis ausencias y celebrar mis pequeños logros como si fueran propios; gracias por ser mujeres ejemplares, modelos a seguir que me inspiran día con día. A Laila, Paola y Ana, quienes a pesar de la distancia, estuvieron presentes durante mi proceso de aprendizaje y siempre tuvieron las palabras de ánimo necesarias para motivarme.

A mis compañeros de clase, por hacer de esta aventura académica una experiencia compartida y llena de momentos inolvidables.

A la institución, por proporcionarme el apoyo necesario y poner en mi camino a docentes que sumaron conocimiento y valor a mi formación profesional día con día.

Finalmente, a la Erika del 2020: hoy puedo mirar hacia atrás con orgullo y decirme a mí misma con total certeza: "Todo valió la pena".

1 Introducción

Contenido

1.1 Antecedentes

En los últimos años, la industria electrónica ha sido una de las industrias con mayor crecimiento a nivel global. Desempeñando un papel fundamental en la innovación y mejora de equipos electrónicos desde teléfonos y computadoras, hasta maquinaria de uso industrial y médico. Todo esto con un solo propósito: mejorar la calidad de vida de las personas.

Los países clave que se han mostrado como líderes en este sector son: Estados Unidos, China, India, Alemania, Brasil y Emiratos Árabes Unidos. Tan solo para enero del 2025 la ESD Alliance [1], una comunidad tecnológica y asociación internacional de empresas de semiconductores, anunció en su reporte Electronic Design Market Data (EDMD) que en el continente americano fue la región con más volumen de ingresos en el Q3 del 2024, por un valor de 2,325.4 millones de dólares.

México también ha fungido como un punto clave en este sector, consolidándose como un actor estratégico gracias a su ubicación geográfica. Su cercanía con Estados Unidos ha incentivado a muchas empresas a invertir en la infraestructura de esta industria. Su posición intermedia permite optimizar la cadena de suministro, facilitando la exportación de productos a gran y mediana escala. Aunado a esto, el talento humano ha sido clave para impulsar la innovación y desarrollo, fortaleciendo la competitividad del país y posicionando a México como una economía emergente en este sector.

[1] SEMI. (2024). Electronic system design industry posts $5.1 billion in revenue in q3 2024, esd alliance reports. URL https://www.semi.org/en/semi-press-release/electronic-system-design-industry-posts-dollar-5.1-billion-in-revenue-in-q3-2024-esd-alliance-reports. Consultado el 6 de septiembre de 2025

Tan solo en el año 2024, en el cuarto trimestre la rama electrónica en México registró un Producto Interno Bruto (PIB) de $6.79 billones de pesos reflejando un aumento del 6.7 % respecto al mismo periodo del año anterior. Según datos del Censo Económico del 2019 se totalizaron 444 unidades económicas en fabricación de componentes electrónicos, destacando Baja California, Jalisco y Chihuahua. Y gracias a esto se mantuvieron 323,000 empleados dependientes a estas unidades económicas, destacando los siguientes municipios con mayor cantidad de empleados: Tijuana (56.9K), Zapopan (42.6K) y Juárez (41.2K) [2].

No obstante, el crecimiento de esta industria se ha visto afectado desde los acontecimientos del COVID-19. La pandemia trajo consigo grandes retos en la cadena de suministro, provocando desabasto de diferentes componentes clave para el cumplimiento de las demandas del mercado, haciendo muy variables los tiempos de entrega y aumentando de manera significativa los precios de la materia prima. Esta alza mantenida y fluctuante en los precios de los materiales continúa sin regularizarse por completo, lo que ha provocado que las empresas busquen adaptarse constantemente para mantener su competitividad. Además, los cambios políticos en las principales potencias y los conflictos geopolíticos han contribuido a prolongar estas tendencias.

Las empresas se encuentran en un proceso continuo para optimizar las estrategias de negociación con proveedores y clientes, con el objetivo de reducir la incertidumbre financiera. En este contexto, se hace evidente la necesidad de herramientas que permitan anticipar las condiciones del mercado en tiempo casi real.

[2] Secretaría de Economía de México. (2019). Fabricación de componentes electrónicos (Data México). URL https://www.economia.gob.mx/datamexico/es/profile/industry/semiconductor-and-other-electronic-component-manufacturing. Consultado el 31 de agosto de 2025

1.2 *Justificación*

Debido a que la industria electrónica es muy amplia, el mercado se ha dividido en dos ramas: por componentes y por aplicaciones. En los componentes podemos encontrar diferentes mercados como lo son: dispositivos de memoria, dispositivos lógicos, sensores, circuitos integrados analógicos, entre otros. Y en la segunda rama podemos encontrar aplicaciones en: la industria automotriz, telecomunicaciones, el procesamiento de datos, la medicina, etc.

En este trabajo, nos enfocaremos en la rama de los componentes, específicamente en los circuitos integrados analógicos (ICs por sus siglas en inglés). Los circuitos integrados o chips analógicos, son circuitos electrónicos que procesan señales análogas de manera continua y son componentes básicos para los dispositivos electrónicos ya que manipulan diferentes señales como temperatura, luz, audio, entre otras. El uso de estos componentes juega un rol muy importante en la era del Internet de las cosas e Inteligencia Artificial, es por eso que las demandas de estos materiales ha aumentado de manera significativa,

trayendo consigo escasez del mismo, lo que hace al producto cada vez más caro.

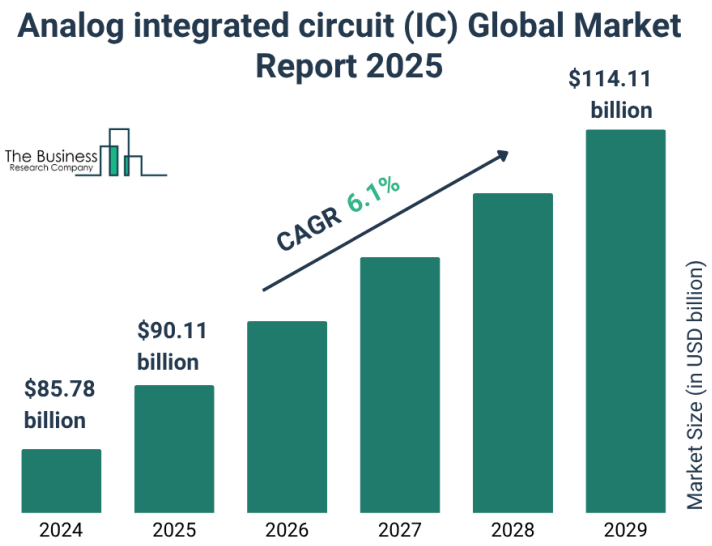

Figura 1.1: *Crecimiento del mercado de los Circuitos Integrados (ICs).* The Business Research Company. (2025). Market Report Image. Tomada de: https://www.thebusinessresearchcompany.com/report/analog-integrated-circuit-ic-global-market-report

La implementación del nowcasting, un término originalmente utilizado en meteorología pero que se ha adaptado ahora a la economía, que se basa en *el aprovechamiento de la valiosa información contenida en un gran número de indicadores del ciclo económico que están disponibles en frecuencias altas- diaria o mensual- para producir estimaciones tempranas de una variable objetivo publicada en una frecuencia menor -trimestral-* [3], podría ser una herramienta útil para anticipar estos cambios.

Con base en la experiencia, es muy normal que los incrementos de precios en los materiales suelen hacerse de manera inesperada, causando un gran impacto en los contratos con clientes. Dado que estos contratos suelen establecerse de manera trimestral o semestral, los clientes esperan una estabilidad que se ve afectada por estos aumentos. Esta situación a menudo resulta en inconformidad que trae como consecuencia el absorber las pérdidas que se puedan generar. Por lo tanto, el nowcasting ofrece la oportunidad de anticipar estos cambios, permitiendo una mejor planificación y negociación que se verá reflejada en la cadena de suministro.

[3] Laura D'Amato, Lorena Garegnani, and Emilio Blanco. Nowcasting de pib: Evaluando las condiciones cíclicas de la economía argentina. Technical Report 2015/69, Banco Central de la República Argentina (BCRA), Investigaciones Económicas (ie), Buenos Aires, 2015

1.3 *Planteamiento del problema*

Los circuitos integrados han mostrado una alta demanda a nivel global y se han visto afectados ampliamente por aumentos significativos

y repentinos de precios, y la falta de visibilidad respecto a estas fluctuaciones representa un desafío importante en la industria, ya que dificulta anticipar escenarios de mercado y limita la capacidad de negociación con clientes y proveedores. La ausencia de mecanismos que permitan prever estas variaciones impactan directamente en la cadena de suministro, afectando la planeación estratégica y reduciendo la competitividad del sector electrónico.

1.4 Hipótesis

La aplicación del nowcasting para pronosticar un precio promedio en los circuitos electrónicos nos ayudará a reducir la incertidumbre en la planificación de costos y optimizar la gestión de la cadena de suministro. Logrando tres objetivos clave: en primer lugar, notificar de manera oportuna y eficaz a los clientes sobre eventuales incrementos de precios; en segundo lugar, fortalecer las negociaciones con nuestros actuales proveedores o tomar la decisión de ampliar el panorama de nuevos socios comerciales; y en tercer lugar, anticipar decisiones de compra que mitiguen el riesgo de desabastecimiento y garanticen la continuidad de la producción.

1.5 Objetivo General

Desarrollar un modelo de nowcasting para estimar las variaciones del precio de los circuitos integrados a corto plazo, con el fin de proporcionar información relevante que nos permita ver las tendencias del mercado.

Para validar si el modelo es funcional, los resultados se compararán con el Índice de Precios al Productor (IPP) de los Circuitos Integrados, que mide el cambio promedio de los precios a lo largo del tiempo. Esta comparación nos permitirá evaluar la precisión del modelo, determinando si las predicciones realizadas se correlacionan de manera efectiva con las tendencias observadas en el IPP.

1.6 Objetivos Específicos

- Explorar diferentes modelos de machine learning que nos ayuden a predecir las variaciones del precio de los circuitos integrados.
- Evaluar y comparar el modelo con las tendencias del IPP.
- A partir de los resultados obtenidos, definir cómo utilizar la información para generar estrategias que nos ayuden a mitigar la incertidumbre financiera y optimizar la gestión de la cadena de suministro.

2 Estado del Arte

El nowcasting es una técnica que inició en el campo de la meteorología, uno de los primeros trabajos relevantes en este campo fue para la predicción de tormentas utilizando datos de radar [1]. Esta capacidad de describir el estado actual y prever cambios en horizontes de tiempo extremadamente cortos permitió que la disciplina permeara eventualmente hacia las ciencias económicas.

La literatura se ha centrado mayoritariamente en el estudio del Producto Interno Bruto (PIB). Ejemplos de ello son los trabajos de Jesús Gálvez-Soriano [2018], quien evalúa modelos para pronosticar el PIB trimestral de México, y Casares [2017], enfocado en la economía de Ecuador. No obstante, ambas investigaciones se basan en métodos econométricos tradicionales, como los Modelos de Factores Dinámicos y Ecuaciones Puente. La adopción de algoritmos de Machine Learning en la literatura económica de nowcasting es aún emergente; si bien existen aplicaciones aisladas con Máquinas de Soporte Vectorial (SVM)[2] o redes neuronales de memoria a corto y largo plazo (LSTM)[3], su implementación para la predicción de índices de precios industriales específicos, como los de la industria electrónica, sigue siendo un área poco explorada.

Piotrowski [2023] destaca que la escasez de materiales críticos en la industria de semiconductores debido a las altas demandas ha comprometido de manera significativa el cumplimiento de la producción y ha disparado volatilidad sin precedentes en los precios de los insumos. Esta inestabilidad creció debido a la pandemia de COVID-19, la cual provocó una disrupción significativa en las cadenas de suministro de manera global. Para comprender la magnitud de esto, se estima que en la industria automotriz se sufrieron pérdidas superiores a $210 billones de dólares[4] debido a la imposibilidad de satisfacer la demanda ante la carencia de semiconductores.

Si bien es cierto, la importancia de poder tener una mejor visión de los precios potencia la competitividad en el mercado, en consecuencia, la implementación de un modelo de nowcasting en la industria electrónica surge como una solución estratégica de vanguardia; al combinar la flexibilidad del Machine Learning con la inmediatez de la

[1] K. Browning and Chris Collier. Nowcasting of precipitating systems. *Reviews of Geophysics - REV GEOPHYS*, 27, 01 1989. DOI: 10.1029/RG027i003p00345

[2] Lei Han, Juanzhen Sun, Wei Zhang, Yuanyuan Xiu, Hailei Feng, and Yinjing Lin. A machine learning nowcasting method based on real-time reanalysis data. *Journal of Geophysical Research: Atmospheres*, 122(7):4038–4051, 2017. DOI: https://doi.org/10.1002/2016JD025783

[3] Xingjian SHI, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun WOO. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015

[4] Aamirah Mohammed and Sardar Asif Khan. Global disruption of semiconductor supply chains during covid-19: An evaluation of leading causal factors. Volume 2: Manufacturing Processes; Manufacturing Systems:V002T06A011, 06 2022

información de alta frecuencia permite anticipar choques de costos y fortalecer la cadena de suministro. Esto resulta crítico para sectores con alta volatilidad, donde la visibilidad en tiempo real de las fluctuaciones a corto plazo es esencial para una planeación estratégica y operativa eficiente.

En la actualidad, las técnicas de Machine Learning se utilizan principalmente para la predicción de la demanda ya sea de manufactureros, distribuidores o clientes, con esto las empresas han minimizado el costo de mantener inventarios y evitar la escasez de diferentes productos[5]. Sin embargo, también se han explorado otras aplicaciones para capturar la complejidad de los mercados globales, Alameer et al. [2019] han demostrado que los modelos de redes neuronales pueden identificar patrones cíclicos y tendencias no lineales, destacando que estos modelos son capaces de reducir el ruido que existe en las series de tiempo financieras. Lago et al. [2021], que centraron sus estudios en el sector energético, demuestran y apoyan la idea de que estos modelos pueden tener un mejor desempeño que los modelos tradicionales.

En este sentido, el Machine Learning representa un cambio fundamental en la capacidad de análisis, enfatizando una gran diferencia entre la predicción y la inferencia causal. Esto se convierte en una relevancia crítica al momento de trasladarse al ámbito operativo; enfocándonos en la cadena de suministro, el uso de técnicas de aprendizaje supervisado permite transformar grandes volúmenes de datos en conocimiento accionable, y como consecuencia se encuentran maneras más inteligentes de aumentar ventas, ahorrar tiempo y resolver problemas. Y a diferencia de los autores Alameer et al. [2019] y Lago et al. [2021], el presente trabajo aborda el desafío de las frecuencias mixtas ya que requiere integrar indicadores de alta frecuencia (diarios) para estimar una variable objetivo de baja frecuencia (mensual, anual o trimestral).

[5] Brahami Menaouer, Fatma Zahra Abdeldjouad, Mohammed Sabri, Khalissa Semaoune, and Matta Nada. Forecasting supply chain demand approach using knowledge management processes and supervised learning techniques. *International Journal of Information Systems and Supply Chain Management*, 15:1–21, 01 2022. DOI: 10.4018/IJISSCM.2022010103

3 Marco teórico

Contenido

En el presente capítulo se abordarán los fundamentos teóricos y metodológicos que serán necesarios para el desarrollo de nuestro modelo de nowcasting.

En primer lugar, se discutirá el desafío que representa el manejo de datos de frecuencia mixta, analizando los métodos econométricos tradicionales, los cuales permiten integrar la información diaria y mensual.

Posteriormente, se explorarán las arquitecturas de Machine Learning, dando un énfasis en las Redes Neuronales, Redes Neuronales de Memoria de Corto y Largo plazo, las Máquinas de Soporte Vectorial y los Bosques Aleatorios, las cuales se han seleccionado ya que proponen una alternativa robusta para capturar las relaciones no lineales.

Finalmente, se describen las técnicas que nos permitirán reducir la dimensionalidad de nuestros datos y las métricas de evaluación que nos ayudarán a medir la precisión y capacidad explicativa de nuestros modelos.

3.1 El desafío de los Datos Mixtos

Como se ha mencionado a lo largo de este trabajo, una de las peculiaridades del nowcasting es la necesidad de integrar datos de diferentes frecuencias. Usualmente, en meteorología, se busca predecir tormentas con datos enfocados en un horizonte de 0 a 6 horas con datos que se obtienen minuto a minuto. En econometría, se busca predecir datos mensuales o trimestrales con información que podemos obtener de manera diaria, o incluso de manera semanal. Esta integración de datos de diferentes frecuencias, conocida como *"El problema de los datos mixtos"*, representa un desafío significativo en la modelización y análisis de series temporales. En este contexto, se han hecho diferentes trabajos y métodos para abordar este problema.

3.1.1 MIDAS

Es común encontrar que la mayoría de los estudios utilizan el método de MIDAS para resolver este problema. Por definición, el Muestreo de Datos Mixtos (MIDAS) nos permite estimar dinámicas que explican una variable de baja frecuencia mediante variables de alta frecuencia y sus rezagos[1]. El modelo original de MIDAS, se puede expresar con la siguiente ecuación:

$$y_t = \beta_0 + \beta_1 \sum_{i=0}^{k} w_i \cdot x_{s(t)-i} + \varepsilon_t$$

[1] Eric Ghysels, Virmantas Kvedaras, and Vaidotas Zemlys-Balevičius. Mixed data sampling (midas) regression models. In Hrishikesh D. Vinod and C.R. Rao, editors, *Financial, Macro and Micro Econometrics Using R*, volume 42 of *Handbook of Statistics*, pages 117–153. Elsevier, 2020. DOI: 10.1016/bs.host.2019.01.005

Donde:

- y_t representa la variable dependiente de baja frecuencia.
- $x_{s(t)-i}$ son los valores rezagados de la variable de alta frecuencia.
- w_i son los pesos que determinan la influencia de cada rezago diario sobre el valor mensual.
- β_0 y β_1 son parámetros del modelo.
- ε_t es el término de error.

Los pesos de nuestro modelo se estiman mediante una función parametrica:

$$w_i = \frac{\psi(\gamma, i)}{\sum_{j=0}^{k} \psi(\gamma, j)}$$

Algunas funciones comunes para ψ incluyen:

- **Beta polinomial:** $\psi(\gamma, i) = x_i^{\gamma_1 - 1}(1 - x_i)^{\gamma_2 - 1}, \quad x_i = \frac{i}{k}$
- **Almon exponencial:** $\psi(\gamma, i) = \exp(\gamma_1 i + \gamma_2 i^2)$

Estas funciones ayudan a encontrar patrones en los datos y controlar los pesos que se le asigna a cada día.

Finalmente, la estimación del modelo se hace mediante mínimos cuadrados no lineales (NLS). El objetivo es minimizar la suma de los errores cuadráticos entre los valores observados y los estimados:

$$\min_{\beta_0,\beta_1,\gamma} \sum_t \left(y_t - \beta_0 - \beta_1 \sum_{i=0}^{k} w_i x_{s(t)-i} \right)^2$$

3.1.2 *Modelos de Factores Dinámicos*

Otra metodología fundamental es la propuesta por Mariano and Murasawa [2003]. A diferencia de MIDAS, este enfoque utiliza modelos de factores dinámicos en un espacio de estados, donde la variable observada de baja frecuencia se considera una media geométrica de las variables latentes (no observadas) de alta frecuencia. De acuerdo con los autores, si definimos $Y_{1,t}$ como un indicador de baja frecuencia observable cada tercer periodo (por ejemplo, trimestral) y $Y^*_{1,t}$ como el nivel mensual latente, la relación se establece mediante la media geométrica de los últimos tres meses:

$$\ln Y_{1,t} = \frac{1}{3}(\ln Y^*_{1,t} + \ln Y^*_{1,t-1} + \ln Y^*_{1,t-2}) \tag{3.1}$$

Esta formulación es particularmente útil porque, al trabajar con logaritmos, la relación se vuelve lineal. Al tomar diferencias de tres periodos para trabajar con tasas de crecimiento, obtenemos la siguiente ecuación que vincula el indicador observado ($y_{1,t}$) con las variaciones mensuales latentes ($y^*_{1,t}$):

$$y_{1,t} = \frac{1}{3}y^*_{1,t} + \frac{2}{3}y^*_{1,t-1} + y^*_{1,t-2} + \frac{2}{3}y^*_{1,t-3} + \frac{1}{3}y^*_{1,t-4} \tag{3.2}$$

Donde $y_{1,t} = \Delta_3 \ln Y_{1,t}$ y $y^*_{1,t} = \Delta \ln Y^*_{1,t}$. Esta estructura permite que, aunque no observemos directamente el dato mensual en ciertos puntos, el modelo pueda estimar su comportamiento como un componente latente influenciado por un factor común (f_t):

$$y^*_t = \mu + \beta f_t + u_t \tag{3.3}$$

3.2 *Machine Learning para series temporales*

Como se mencionó anteriormente, los modelos tradicionales de series temporales asumen una linealidad que puede resultar insuficiente ante la volatilidad extrema y los choques estructurales propios de la industria de semiconductores, dichas relaciones pueden presentar patrones no lineales que los métodos clásicos no logran capturar.

Es por eso que el presente trabajo propone un enfoque híbrido basado en modelos de Machine Learning para aplicar el Nowcasting. Esta metodología busca superar las limitaciones de linealidad, permitiendo identificar dinámicas más complejas en los precios de los semiconductores, logrando así un modelo más robusto. El modelo integra la media geométrica como mecanismo de agregación para resolver el problema de los datos mixtos, ya que esta es base en la metodología propuesta por Mariano y Murasawa[2]. Asimismo, se retoma la lógica de los modelos de MIDAS al incorporar una estructura de rezagos en las variables de alta frecuencia; permitiendo que los algoritmos capturen la dinámica temporal de la información pasada.

[2] Roberto S. Mariano and Yasutomo Murasawa. A new coincident index of business cycles based on monthly and quarterly series. *Journal of Applied Econometrics*, 18(4):427–443, None 2003. DOI: 10.1002/jae.695

3.2.1 *Redes Neuronales*

Las redes neuronales son modelos de regresión no lineales inspirados en la estructura del cerebro humano. La unidad básica es la "neurona" o nodo, la cual recibe múltiples entradas, le asigna un peso específico y con ayuda de una función de activación genera una salida. La peculiaridad de este modelo es que puede aprender representaciones jerárquicas y no lineales.

3.2.2 *Redes Neuronales de Memoria a Corto y Largo Plazo (LSTM)*

Las Redes Neuronales de Memoria a Corto y Largo Plazo, es un tipo de red neuronal recurrente. donde durante el entrenamiento se utiliza información secuencial que viaja a través de la red neuronal desde el vector de entrada hasta las neuronas de salida, mientras que el error se calcula y se propaga de vuelta a través de la red para actualizar parámetros. La información en estas redes incorpora bucles en la capa oculta, estos permiten que la información fluya multidireccionalmente, de modo que el estado oculto representa la información pasada contenida en un intervalo de tiempo determinado. En consecuencia, la salida depende de las predicciones previas ya conocidas[3]. Las redes LSTM introducen una unidad de memoria y un mecanismo de compuerta para permitir la captura de las largas dependencias en una secuencia, por lo tanto, pueden recordar u olvidar información selectivamente y son capaces de aprender miles de pasos de tiempo mediante estructuras llamadas estados de celda y tres compuertas:

[3] Hayrettin Okut. *Deep Learning: Long-Short Term Memory*. 06 2021

- Compuerta de olvido: decide qué información de estados anteriores debe descartar.
- Compuerta de entrada: Determina qué nueva información se almacena en el estado de la celda.

- Compuerta de salida: Controla qué información se transmite a la siguiente capa.

Esta arquitectura resulta ideal para el nowcasting, ya que permite que el modelo recuerde la dinámica de los rezagos de alta frecuencia de manera selectiva.

Diseño, arquitectura e hiperparámetros. Para crear la estructura de una red neuronal, es importante saber que existen diferentes parámetros e hiperparámetros que pueden ajustarse para el funcionamiento óptimo de la red[4].

[4] Keras Team. Model training APIs. https://keras.io/api/models/model_training_apis/, 2023. Accedido el: 16 de febrero de 2026

- **Optimizador:** Estos son algoritmos que ayudan a encontrar los pesos óptimos para minimizar la función de perdida en nuestra red. Para este trabajo, utilizaremos los siguientes[5]:

[5] Mike Bernico. *Deep Learning Quick Reference: Useful hacks for training and optimizing deep neural networks with TensorFlow and Keras*. Packt Publishing Ltd., 2018

  - **Gradiente Descendiente con Momento:** es una variante del gradiente descendiente, pero su diferenciador es que incrementa la rapidez del aprendizaje hacia la dirección donde el gradiente ha sido constante y reduce la tasa de aprendizaje en las direcciones de alta fluctuación. Utiliza la media móvil ponderada de los gradientes previos, lo cual ayuda a que el algoritmo incremente su velocidad de aprendizaje de manera estable hacia el mínimo global.

$$v_t = \beta v_{t-1} + (1-\beta) grad^2 \tag{3.4}$$

$$\theta = \theta - v_t \tag{3.5}$$

  - **RMSProp:** es un algoritmo avanzado diseñado para acelerar la convergencia del descenso de gradiente. Su función se basa en ajustar las tasas de aprendizaje para cada parámetro, permitiendo un avance más veloz en dimensiones con gradientes consistentes y amortiguando oscilaciones en aquellas direcciones donde el gradiente tiene cambios significativos en los pesos de la red.

$$grad = \frac{\partial J}{\partial \theta} \tag{3.6}$$

$$v_t = \beta v_{t-1} + (1-\beta) grad^2 \tag{3.7}$$

$$\theta = \theta - \alpha \frac{grad}{\sqrt{v_t}} \tag{3.8}$$

  - **Adam:** combina las mejores partes del descendiente del gradiente con momento y RMSProp. Logra estabilizar el entrenamiento,

acelerando la convergencia.

$$grad = \frac{\partial J}{\partial \theta} \tag{3.9}$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) grad \tag{3.10}$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) grad^2 \tag{3.11}$$

$$\theta = \theta - \alpha \frac{m_t}{\sqrt{v_t} + \epsilon} \tag{3.12}$$

- **Tasa de aprendizaje:** es un hiperparámetro que ayuda a garantizar que un modelo aprenda lo suficiente en el entrenamiento, tiene que tener el equilibrio preciso para permitir que la red aprenda lo suficientemente rápido para avanzar de manera considerable, pero no demasiado como para que el aprendizaje sea inestable y tenga oscilaciones caóticas que impidan la convergencia.
- **Épocas:** se refiere al ciclo completo de entrenamiento de la red con todos los datos disponibles, implica que cada dato de la muestra ha tenido la oportunidad de actualizar sus pesos.
- **Batch size:** es un hiperparámetro que define la cantidad de muestras procesadas antes de que los parámetros internos del modelo (pesos) se actualicen durante el entrenamiento.

3.2.3 *Árboles de Decisión, Bolsa de Árboles y Bosques Aleatorios*

Los árboles de decisión son modelos que dividen el espacio de los datos en regiones mediante reglas binarias de decisión. El mayor problema de estos, es que tienden al sobreajuste, es por eso que se utilizan los Bosques Aleatorios o las Bolsas de Árboles, que son un conjunto de múltiples árboles de decisión entrenados con diferentes subconjuntos de datos. La predicción final es el promedio de las salidas de todos los árboles, lo que reduce la varianza y mejora la robustez del modelo ante la volatilidad.

Diseño, arquitectura e hiperparámetros. Al igual que las redes neuronales, para asegurar el funcionamiento óptimo de los árboles de decisión es necesario ajustar los diferentes hiperparámetros[6,7,8]

- **Splits:** es el número de subgrupos en los cuales los datos se dividen.
- **Hojas:** son los nodos finales del árbol, cuando se llega al número establecido ya no se realizan más divisiones.
- **Profundidad:** indica el límite de niveles que puede tener el árbol desde la raíz hasta la hoja más lejana y nos ayuda a regular la complejidad del modelo.

[6] Scikit-learn: Decision tree regressor. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html, 2011d. Accedido el: 16 de febrero de 2026

[7] Scikit-learn: Bagging regressor. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingRegressor.html, 2011c. Accedido el: 16 de febrero de 2026

[8] Scikit-learn: Random forest regressor. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html, 2011e. Accedido el: 16 de febrero de 2026

- **Número de estimadores:** es la cantidad de árboles que conforman nuestro bosque o bolsa
- **Muestras máximas:** determinan el porcentaje de datos de la muestra que se utilizará para entrenar cada árbol.

3.2.4 *Máquinas de Soporte Vectorial (SVM)*

Las Máquinas de Soporte Vectorial son modelos que se caracterizan por ser robustos, estos tienen como objetivo encontrar un hiperplano óptimo que minimice el error dentro de un umbral específico. Una de las mayores ventajas de este modelo es el uso de los núcleos (Kernel), los cuales nos permiten proyectar los datos en un espacio de mayor dimensión donde las relaciones no lineales pueden ser tratadas como lineales. Lo cual es particularmente conveniente para el nowcasting, ya que ofrece una alta capacidad para tratar valores atípicos.

Diseño, arquitectura e hiperparámetros. Para las máquinas de soporte vectorial existen cuatro hiperparámetros principales[9]:

- **C:** este hiperparámetro nos ayuda a lograr un equilibrio entre el margen de decisión y nos ayuda a minimizar el error, pero nos puede llevar al sobreajuste con valores muy altos.
- **Kernel:** es una función matemática que nos ayuda a transformar la información en un espacio de mayor dimensión que nos ayuda a hacer una separación lineal de la información.
- **Gamma:** Controla el alcance de la influencia de las observaciones individuales.
- **Epsilon:** Establece el margen de tolerancia ante errores de predicción pequeños.

[9] Scikit-learn: Svr. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html, 2011b. Accedido el: 16 de febrero de 2026

3.3 *Reducción de Dimensionalidad y Extracción de Características*

Dado que a menudo se cuenta con una gran cantidad de variables en los modelos, se considera fundamental la aplicación de técnicas para reducir la dimensionalidad y extraer solo aquellas características relevantes.

3.3.1 *Análisis de Componentes Principales (PCA)*

El PCA es una técnica estadística que nos ayuda a transformar nuestras variables dependientes que tengan una alta correlación en un número menor de variables no correlacionadas llamadas componentes principales. El objetivo de esta técnica es capturar la mayor cantidad de

varianza posible de los datos originales, permitiendo reducir el ruido y la complejidad en los modelos de machine learning sin perder la escencia de la información.

3.3.2 *Mínimos Cuadrados Parciales (PLS)*

A diferencia del PCA, esta técnica busca encontrar nuevas variables latentes que maximicen la varianza entre la variable dependiente y las variables independientes, al mismo tiempo que reduce la dimensionalidad. Esta técnica es particularmente útil cuando se tienen muchas variables independientes que están altamente correlacionadas, lo que es común en el nowcasting debido a la gran cantidad de indicadores económicos disponibles.

3.4 *Métricas de evaluación*

Para evaluar el desempeño de los diferentes modelos implementados se utilizarán dos métricas fundamentales en el análisis de la regresión:

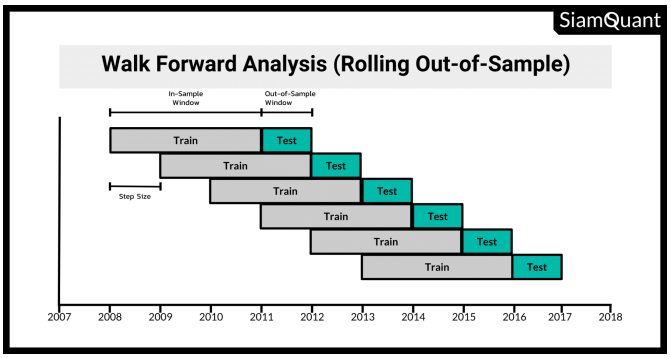

Figura 3.1: Ejemplo gráfico de la evaluación a través del tiempo con la prueba RWF. Tomada de SiamQuant https://www.siamquant.com/walk-forward-analysis/.

- **Error Cuadrático Medio (MSE):** mide el promedio de los errores al cuadrado entre los valores observados y las predicciones del modelo, y penaliza de manera severa las desviaciones grandes, un valor cercano a cero indica una mayor precisión.
- R^2**:** mide la proporción de la varianza de la variable dependiente que es explicada por las variables independientes del modelo, un valor cercano a 1 indica un mejor ajuste del modelo.
- **Rolling Walk-Forward Validation:** es una metodología para series de tiempo ya que utiliza la dependencia temporal de los datos. Consiste en desplazar una ventana de entrenamiento a lo largo del tiempo y se valida con el periodo posterior; después la ventana de entrenamiento se desplaza un paso hacia adelante incorporando los nuevos datos recién observados, así como se muestra en la figura 3.1.

- **Probabilidad de Sobreajuste en el Backtest (PBO):** se define como la probabilidad condicional de que la estrategia o configuración del modelo seleccionado como óptimo durante el periodo de prueba (dentro de la muestra o In-Sample) termine teniendo un rendimiento inferior a la mediana de todas las estrategias alternativas cuando se evalúa en datos nuevos (fuera de la muestra y Out-of-Sample). La PBO cuantifica exactamente *"qué tan probable es que el aparente éxito de un backtest sea simplemente un falso positivo."* Para calcular empíricamente la Probabilidad de Sobreajuste en Backtest (PBO), el marco de trabajo propone un algoritmo computacional denominado Validación Cruzada Simétrica Combinatoria (CSCV). El procedimiento para estimar la PBO se divide en cinco etapas fundamentales, siguiendo la propuesta Bailey et al. [2017]:

  1. **Construcción de la matriz de rendimientos (M):** Se forma una matriz $M \in \mathbb{R}^{T \times N}$ recolectando el histórico de rendimiento de todas las pruebas realizadas, donde:
     - Las columnas (N) representan cada una de las configuraciones o alternativas de la estrategia probadas.
     - Las filas (T) representan las observaciones temporales sincrónicas para todas las columnas.
  2. **Partición de datos en submatrices:** Se divide la matriz M horizontalmente por filas en un número par (S) de submatrices disjuntas de idéntico tamaño. Esto preserva la dependencia temporal y estacionalidad dentro de cada bloque. Se recomienda comúnmente un valor de $S = 16$.
  3. **Generación de combinaciones simétricas (C_S):** Se agrupan las submatrices en todas las combinaciones posibles conformadas por exactamente la mitad de los bloques ($S/2$). Para $S = 16$, esto genera $\binom{16}{8} = 12,870$ escenarios de prueba.
  4. **Entrenamiento y prueba sistemática (IS vs. OOS):** Para cada combinación generada, se itera el siguiente proceso:
     - **División de la muestra:** Se unen los $S/2$ bloques de la combinación para formar el conjunto de entrenamiento o *In-Sample* (J). Los bloques restantes conforman el conjunto de prueba u *Out-Of-Sample* ($\bar{J}$).
     - **Medición del rendimiento:** Se calcula la métrica de desempeño (e.g., ratio de Sharpe o MSE) para las N estrategias en ambos conjuntos (J y $\bar{J}$).
     - **Identificación del ganador:** Se elige la estrategia con mejor rendimiento dentro del conjunto J.
     - **Rastreo del rango relativo ($\bar{\omega}_c$):** Se determina la posición relativa que obtuvo dicha estrategia ganadora dentro del

conjunto $\bar{J}$ frente al resto de las alternativas.

- **Cálculo del logit (λ_c):** Se transforma el rango en un logit mediante la expresión:

$$\lambda_c = \ln\left(\frac{\bar{\omega}_c}{1-\bar{\omega}_c}\right)$$

el cual indica la consistencia del desempeño entre las muestras.

5. **Estimación de la PBO final:** Al finalizar las iteraciones, se obtiene una distribución de logits. La PBO se define como la proporción de iteraciones en las que la estrategia óptima *In-Sample* obtuvo un rendimiento por debajo de la mediana en el conjunto *Out-Of-Sample* ($\lambda_c > 0$). Un valor de PBO $> 0{,}05$ sugiere que el éxito del modelo podría deberse al ajuste del ruido histórico.

- **Bootstrap Reality Check:** [10] es una prueba de hipótesis diseñada para abordar el problema de *"data snooping"*, con la técnica de remuestreo evalúa si el modelo con mejor rendimiento entre los otros modelos posee superioridad o si sus resultados se deben al azar.

  Tiene como base la **hipótesis nula (H_0)**:

  "El mejor modelo evaluado durante la búsqueda no tiene ninguna superioridad predictiva frente al modelo de referencia."

  Se proponen dos enfoques:

  - **Monte Carlo Reality Check:** Calcula un estimador consistente de la matriz de covarianza de los modelos y luego extrae muestras repetidas de una distribución normal multivariada para obtener los percentiles.
  - **Bootstrap Reality Check:** Utiliza el método de *stationary bootstrap*, el cual está adaptado para datos de series temporales dependientes típicos de los mercados financieros. Agrupa los datos en bloques de longitud aleatoria, para preservar la dependencia temporal de los datos.

  En este trabajo de investigación, se propone utilizar el **Stationary Bootstrap** y sigue los siguientes pasos:

  1. **Definición:** Se define el modelo base (*benchmark*), los modelos candidatos y se elige la cantidad de remuestreos (por ejemplo, $N = 500$ o 1000).
  2. **Remuestreo:** Se generan N índices aleatorios de remuestreo sobre la serie temporal histórica.
  3. **Evaluación recursiva:** Se evalúa cada modelo recursivamente: se calcula el rendimiento del modelo frente al *benchmark* con los datos originales, y luego con las N muestras generadas por el *bootstrap*.

[10] Halbert White. A reality check for data snooping. *Econometrica*, 68(5):1097–1126, 2000. DOI: https://doi.org/10.1111/1468-0262.00152. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/1468-0262.00152

4. **Registro de rendimientos:** Se guarda el rendimiento máximo alcanzado tanto en los datos originales como en las simulaciones de *bootstrap*.
5. **Cálculo del P-valor:** El *Reality Check p-value* final se obtiene comparando el rendimiento máximo original con los percentiles de la distribución de los rendimientos máximos simulados.

Al combinar estas métricas podremos realizar una evaluación completa, equilibrando la magnitud de los errores residuales con la capacidad explicativa de nuestro modelo.

4 Desarrollo Metodológico

Contenido

En el presente capítulo se describe de manera detallada la metodología utilizada para la construcción y validación de nuestros modelos. Como se comentó anteriormente, el propósito de esta tesis es la realización del nowcasting para estimar los cambios de precios en los circuitos integrados a corto plazo. Para lograrlo, el proceso de investigación se inició con la identificación de nuestra variable dependiente y nuestras variables independientes, dando prioridad a indicadores que presenten una alta correlación teórica para después demostrar su relevancia estadística.

La base de datos consolidada y los archivos de código generados para esta investigación están disponibles a solicitud del lector para fines académicos.

4.1 Selección de variables

Índice de Precios del Productor (IPP). El Índice de Precios del Productor (IPP) es un indicador que mide de manera mensual las variaciones de los precios de los bienes producidos y vendidos por los productores de un país y permite conocer si los precios de los bienes que se fabrican en una economía están subiendo o bajando, actuando como un indicador de la inflación.

Dicho lo anterior, para nuestro trabajo de estudio, se pretende predecir el IPP de los semiconductores tomando como base las estadísticas de The Bureau of Labor Statistics (BLS) de los Estados Unidos, que es la agencia encargada de medir la actividad comercial, condiciones laborales y los cambios de precios en la economía[1]. Al medir los cambios de precios desde la perspectiva del productor (en este caso los manufactureros de la industria electrónica), este índice nos ayudará a identificar presiones en los costos de los insumos antes de que se vean reflejados en nuestro producto final, convirtiéndose en una métrica esencial para la gestión de riesgos.

[1] U.S. Bureau of Labor Statistics. URL https://www.bls.gov/ppi/

Materias primas de los circuitos integrados. Dado que el objetivo principal del nowcasting radica en aprovechar la información de alta frecuencia, se seleccionaron como variables independientes aquellos insumos críticos para la construcción de los semiconductores, cuya dinámica en el mercado puede anticipar las fluctuaciones en el costo de los semiconductores. Bajo este criterio, se seleccionaron metales preciosos o industriales, así como energéticos, cuya escasez o encarecimiento impactan de manera directa en el margen de producción de los semiconductores. La extracción de las series históricas diarias se obtuvieron mediante la plataforma FactSet, una plataforma global de gestión de datos y análisis financiero utilizada por instituciones académicas, ya que proporciona el acceso instantáneo a datos financieros y análisis que los inversores utilizan para tomar decisiones cruciales[2]. Los metales que se eligieron fueron los siguientes:

[2] Factset website. https://www.factset.com/

- **Plata y Cobre:** Esenciales por su alta conductividad eléctrica; el cobre es el estándar para marcos de plomo e interconexiones, mientras que la plata se utiliza en pastas conductoras y contactos.
- **Oro:** Utilizado predominantemente en los hilos de conexión debido a su excepcional resistencia a la corrosión y maleabilidad.
- **Silicio y Paladio:** El silicio representa el sustrato base (oblea) de la industria, mientras que el paladio es crítico para la fabricación de condensadores cerámicos multicapa (MLCC) y procesos de galvanoplastia.
- **Petróleo:** Actúa como un indicador de los costos logísticos y de energía, además de ser la materia prima básica para los polímeros utilizados en el encapsulado de los componentes.

A pesar de tener una buena fuente como Facset para obtener nuestra información, la serie temporal del Silicio no estaba disponible, por lo tanto se decidió usar como variable Proxy el Silicio de Magnesio.

Variables macroeconómicas. Las dinámicas que existen en la economía global son un factor determinante ya que influyen de manera directa en las expectativas cambiantes del mercado y en los costos de las transacciones entre países. Bajo esta premisa, se buscó incorporar las tasas de cambio como una variable de alta frecuencia para expresar cómo se comporta la oferta y demanda entre los bloques comerciales. Específicamente, se seleccionó el tipo de cambio entre el Dólar estadounidense (USD) y el Yuan chino (CNY), dado que el dólar es una unidad dominante en el comercio internacional y el Yuan chino tiene una posición importante en la manufactura y ensamble de semiconductores a nivel mundial. Esta variable nos permitirá que el modelo capture el impacto de tensiones comerciales y las fluctuaciones monetarias en el costo final de los circuitos integrados. La información fue capturada de FRED, una base de datos pública que centraliza miles de series temporales de indicadores económicos regionales, nacionales e internacionales[3].

[3] Chinese yuan renminbi to u.s. dollar spot exchange rate. https://fred.stlouisfed.org/series/DEXCHUS

4.2 *Análisis exploratorio de datos*

Como se mencionó en la sección previa, se pretende formar nuestro modelo con siete variables independientes de alta frecuencia y una variable dependiente (IPP), su evolución histórica y comportamiento lo podemos ver de manera gráfica en la Figura 4.1.

Como es de esperarse, las series temporales recolectadas presentan diferentes horizontes de información, por lo que se requirió un primer análisis para determinar el periodo de estudio. Como se puede observar en la Tabla 4.1, a pesar de que la mayoría de nuestras variables de alta frecuencia cuentan con registros desde febrero de 2006, la serie correspondiente al Silicio presenta una restricción de inicio en septiembre del 2014, actuando como una limitante para la construcción de nuestra base de datos. Por otro lado, nuestra variable objetivo (IPP) dispone información hasta diciembre del 2025, con lo cual se establece un límite superior de los datos observados para el entrenamiento y validación de nuestros modelos.

Variable	Fecha de Inicio	Fecha de Finalización
IPP	01-06-2005	01-12-2025
Plata	21-02-2006	20-02-2026
Cobre	21-02-2006	20-02-2026
Oro	21-02-2006	20-02-2026
Silicio	29-09-2014	13-02-2026
Paladio	21-02-2006	20-02-2026
Petróleo	21-02-2006	20-02-2026
Tasa de cambio Yuan/Dolar	02-01-1981	13-02-2026

Cuadro 4.1: Periodos de disponibilidad de las series de tiempo.

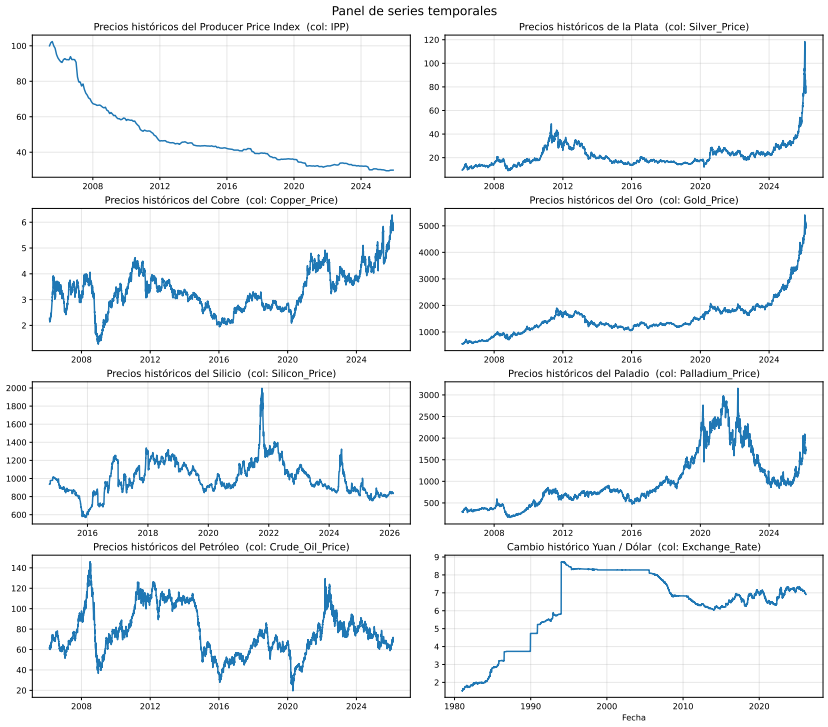

Figura 4.1: *Series temporales de nuestras variables de estudio*

En consecuencia a lo anterior dicho, se decidió eliminar la variable Proxy del Silicio y delimitar nuestro horizonte de estudio entre el 21 de febrero del 2006 y 1 de diciembre del 2025.

4.3 *Transformación de datos y tratamiento de datos faltantes*

Después de haber determinado nuestro periodo de análisis, se procedió con la consolidación y pre-procesamiento de nuestras variables independientes de alta frecuencia. Durante esta fase, se identificaron registros nulos debido a períodos de inactividad bursátil (fines de semana); como medida inicial se optó por eliminar aquellas observaciones donde la totalidad de los indicadores carecían de información. Esta decisión busca mantener la calidad de nuestra información y evitar introducir sesgos o ruidos artificiales. Posteriormente, se realizó un análisis detallado para determinar si aún existían datos faltantes y seleccionar un tratamiento óptimo para ellos, los resultados se presentan en la Tabla 4.2

Debido a que el porcentaje de datos faltantes resultó ser bajo, ya que no supera el 5 % en ninguna de las variables, se eligió un proceso de imputación estadística para tratarlos. El método seleccionado fue la interpolación lineal, ya que esta tiene la capacidad de preservar la

Variable	# Datos faltantes	% Datos faltantes
Tasa de cambio Yuan/Dolar	169	3.2 %
Plata	152	2.9 %
Cobre	150	2.9 %
Oro	149	2.9 %
Paladio	149	2.9 %
Petróleo	23	0.44 %

Cuadro 4.2: Porcentaje de datos faltantes.

tendencia local sin generar ruidos artificiales como lo podrían hacer algunos otros métodos. La interpolación lineal asegura una transición suave entre los puntos observados, manteniendo la estructura de nuestros datos.

Posterior al tratamiento de datos faltantes, se procedió con la transformación de las variables a una periodicidad mensual, esto con el objetivo de alinear la información con nuestra variable objetivo (IPP). Como se mencionó anteriormente, para este proceso se implementó una estructura de ventana de rezagos (lags) basada en la media geométrica. Esta técnica de agregación nos permite promediar tasas de variación y mitigar el impacto de valores extremos sin distorsionar las tendencias de la serie.

El proceso fue el siguiente, por cada mes t, se estructuró un vector de rezagos conformado por las observaciones diarias de los 28 días inmediatamente anteriores al cierre del mes de referencia. Sobre este vector se aplicó una transformación mediante la media geométrica para colapsar la dimensionalidad de los datos diarios en un único valor representativo mensual por predictor. La fórmula de esta transformación se expresa de la siguiente manera:

$$\bar{x}_{geom,t} = \left(\prod_{i=1}^{n} x_{t-i} \right)^{1/n} \tag{4.1}$$

donde $n = 28$ representa el número de observaciones diarias consideradas para capturar la dinámica de precios previa al reporte del IPP.

Como resultado de esta transformación temporal, la base de datos definitiva queda construida por una matriz de seis variables independientes y una variable objetivo, ambas con una frecuencia mensual. Y en total el conjunto de datos final contará con 237 observaciones.

Transformación de datos para la red neuronal LSTM Para el caso específico de nuestra red neuronal, se omitió la agregación mediante la media geométrica ya que esta red nos permite preservar la secuencia temporal diaria. En su lugar, se desarolló una función para extraer ventanas temporales que capturan los 28 días previos a cada

observación mensual del IPP, transformando la base de datos en una matriz tridimensional con las siguientes dimensiones (237,28,6). Este diseño nos permite que la red analice de manera directa la volatilidad diaria y las tendencias de toda la información.

4.4 *Selección de variables*

Antes de proceder con la exploración de la distribución y normalización de nuestros datos, se realizó un análisis de relevancia estadística para validar la capacidad explicativa de los indicadores seleccionados. La primera herramienta utilizada para este diagnóstico fue el coeficiente de correlación de Pearson, el cual lo expresamos y validamos a través de una matriz de calor como se muestra en la Figura 4.2. A partir de esta, se identificó que los precios del Oro y el Paladio presentan las correlaciones negativas más fuertes frente a nuestra variable objetivo, seguidos de la Plata y el Tipo de Cambio. Por el contrario, los precios del Cobre y el Petróleo mostraron asociaciones débiles, sugiriendo que no tienen influencia sobre el IPP.

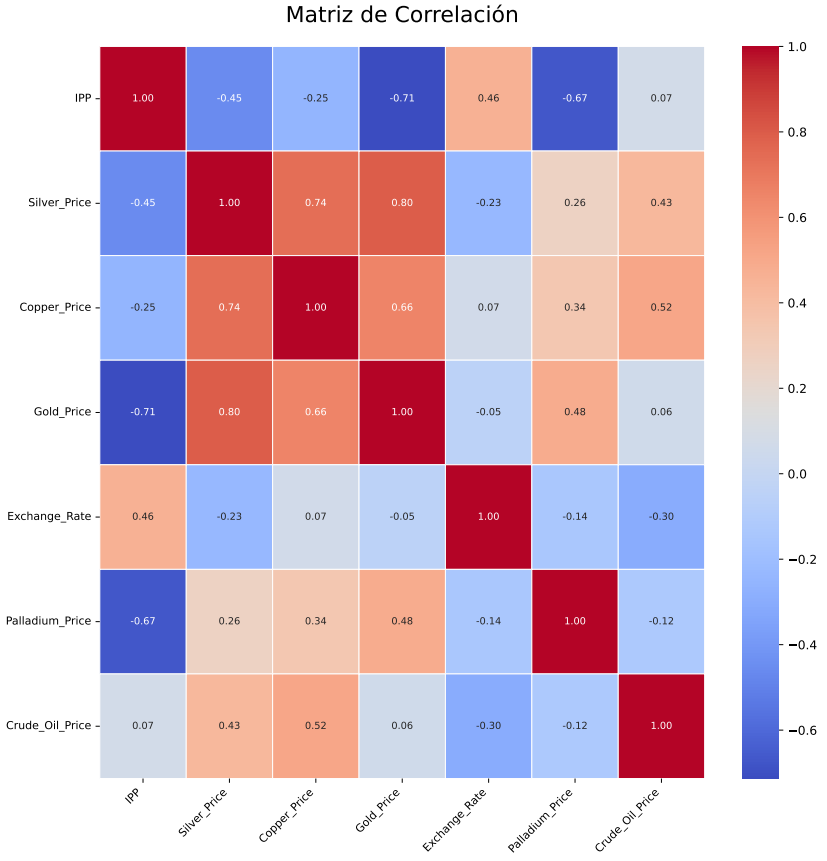

Figura 4.2: *Matriz de correlación.*

Con el objetivo de validar la relevancia individual de cada variable, se estimó un modelo de regresión lineal para evaluar la significancia estadística con ayuda de los **p-valores**. Utilizando un nivel de confianza del 95 %, se buscó contrastar la hipótesis nula, la cual asume la ausencia de evidencia o efecto significativo entre variables y sugiere que los resultados se deben al azar. Como se puede observar en la Tabla 4.3, la mayoría de nuestros resultados presentan un p-valor de 0.00, con lo cual podemos rechazar nuestra hipótesis nula. Sin embargo, el Petróleo obtuvo un p-valor de 0.217, lo que confirma que no existe evidencia para mostrar que cuenta con significancia estadística dentro de este modelo.

Variable	**P-valor ($P > \|t\|$)**
Plata (Silver_Price)	0.000
Cobre (Copper_Price)	0.000
Oro (Gold_Price)	0.000
Tipo de Cambio (Exchange_Rate)	0.000
Paladio (Palladium_Price)	0.000
Petróleo (Crude_Oil_Price)	0.217

Cuadro 4.3: Análisis de significancia estadística (P-valores) por variable.

Como última prueba, se realizó un análisis de la importancia de nuestras variables con ayuda de un modelo de Árbol de decisión. Éste reveló (Figura 4.3) que el precio del Paladio predomina con una importancia superior al 80 %, seguido por el tipo de cambio. Resulta notable que, a pesar de la relevancia teórica, la variable del Petróleo muestra una contribución casi igual al 0 %. Lo cual es consistente con el valor que obtuvimos en el p-valor de nuestra regresión lineal.

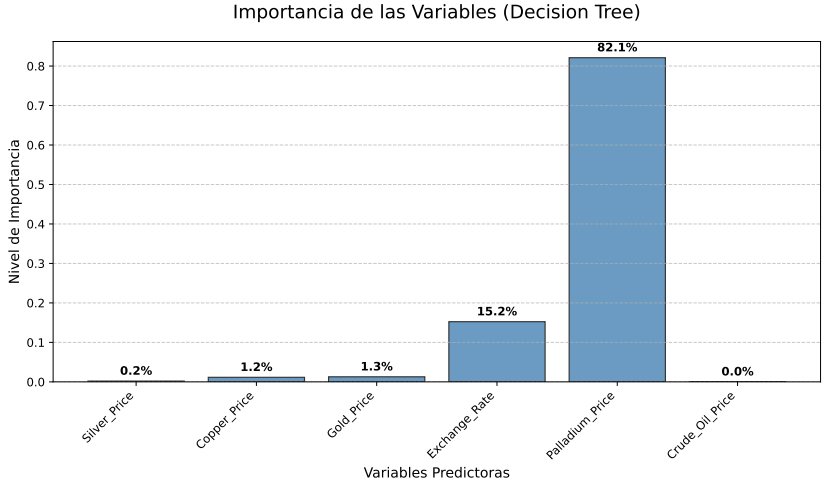

Figura 4.3: *Nivel de importancia de las variables.*

Es por eso que de acuerdo a los diferentes resultados obtenidos, se decidió eliminar la variable del Petróleo de nuestro estudio, ya que no

aporta información al modelo y podría causar un ruido innecesario que afectaría en el entrenamiento. Esto nos deja con un total de cinco variables independientes.

4.5 *Normalización.*

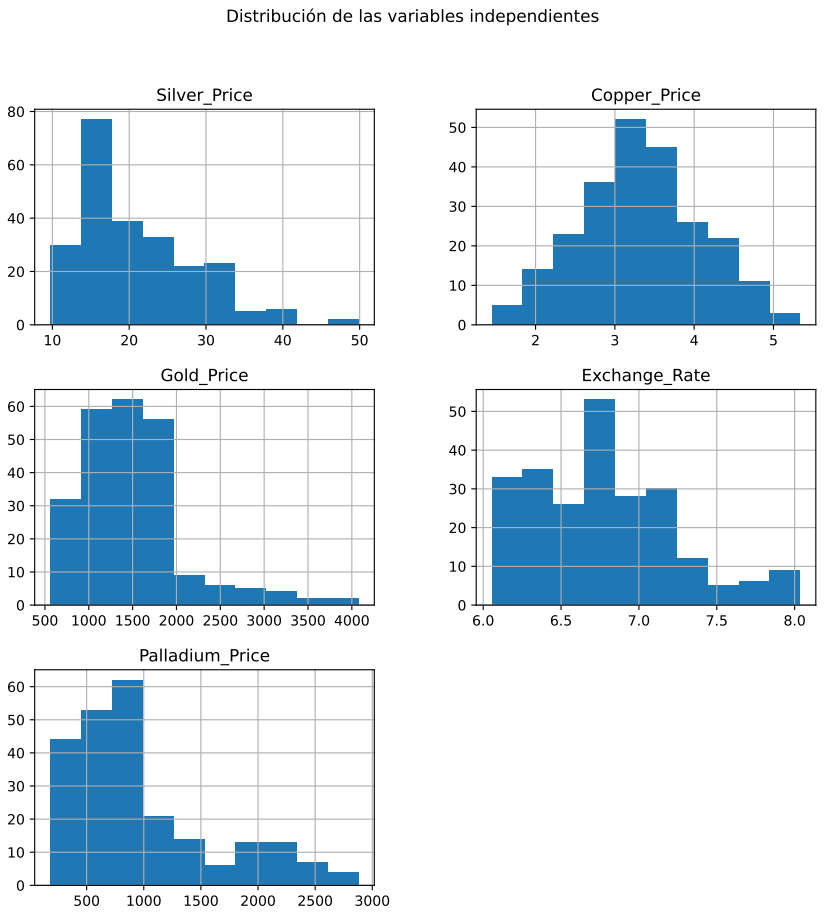

Figura 4.4: *Distribución original de las variables independientes.*

Para la etapa final del pre-procesamiento, verificamos la distribución estadística de las variables, dado que un sesgo excesivo puede comprometer la capacidad del aprendizaje de los modelos de machine learning. Podemos observar en la Figura 4.5 que los indicadores como la Plata, el Oro y el Paladio presentan un sesgo positivo a la derecha, lo cual indica que una corrección en los datos nos ayudaría a estabilizar la varianza.

Original	**Box-Cox**	**Log**	**Sqrt**
6	5	8	7

Cuadro 4.4: Frecuencia de valores atípicos detectados por método de transformación.

Paralelamente, se realizó una prueba para la detección de valores atípicos con el fin de evitar introducir ruido en el entrenamiento. Para ello, se empleó la distancia de Mahalanobis, la cual, a diferencia de la distancia Euclidiana, no asume independencia entre las variables; por el contrario, utiliza la matriz de covarianza para identificar anomalías en los datos altamente correlacionados, como es el caso del precio de los metales y el IPP. Bajo este criterio, se identificaron originalmente 8 datos atípicos, como se resume en la Tabla 4.4.

Con el fin de normalizar la información y reducir el número de anomalías, se evaluaron tres transformaciones para comparar sus efectos sobre el sesgo, la curtosis y la frecuencia de los outliers.

Los resultados de las Tablas 4.5 y 4.6 confirman que la transformación de Box-Cox es la más robusta para reducir significativamente estas medidas estadísticas de la distribución y a la vez proporcionan un conjunto de datos con menos datos atípicos.

Variable	Original	Box-Cox	Log	Sqrt
Silver_Price	1.0859	0.0325	0.3574	0.7090
Copper_Price	0.0868	-0.0199	-0.5791	-0.2236
Gold_Price	1.5089	-0.0005	0.0160	0.7340
Exchange_Rate	0.6625	0.0488	0.5015	0.5810
Palladium_Price	1.1271	-0.0052	-0.0724	0.5913

Cuadro 4.5: Comparativa de niveles de sesgo por método de transformación.

Variable	Original	Box-Cox	Log	Sqrt
Silver_Price	1.0234	-0.5671	-0.4751	0.0200
Copper_Price	-0.3492	-0.3075	0.4753	-0.1314
Gold_Price	3.5872	0.3737	0.3825	1.3341
Exchange_Rate	0.1179	-0.7046	-0.1720	-0.0350
Palladium_Price	0.4332	-0.5119	-0.4851	-0.3520

Cuadro 4.6: Impacto de las transformaciones en el nivel de curtosis.

4.6 *Segmentación y estandarización de datos.*

Con el fin de evaluar la capacidad y robustez de nuestros modelos, se diseñaron dos esquemas de segmentación de datos. En ambos escenarios se reservaron las últimas 6 observaciones mensuales (los cuales corresponden al período final del 2025) con el fin de realizar una validación cruzada. Sobre el conjunto de datos remanente, se ejecutaron las siguientes variaciones experimentales:

- **Prueba A (Temporal):** División secuencial 80 % entrenamiento y 20 % prueba, preservando el orden cronológico (Figura 4.5).
- **Prueba B (Aleatoria):** División al azar usando el 80 % para entrenamiento y 20 % de prueba.

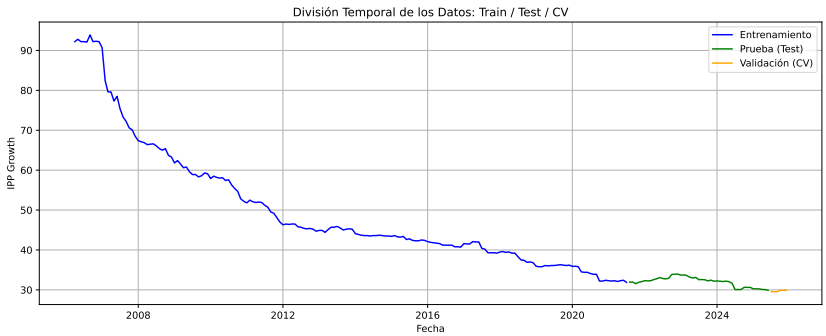

Figura 4.5: *Prueba A: División temporal de nuestra variable dependiente para el entrenamiento de nuestro modelo.*

Estandarización de Variables Como etapa final previa a la implementación de los modelos de machine learning, se procedió al escalamiento de nuestras variables independientes mediante la técnica de estandarización (StandardScaler), la cual transforma los datos para que presenten una media de cero y una desviación estándar de uno. Esto facilita la convergencia de los algoritmos de optimización y asegura que ninguna carácterística domine en el aprendizaje debido a su magnitud y nos ayuda a evitar el problema de alta colinealidad.

4.7 *Implementación de modelos de Machine Learning.*

Después de haber finalizado la etapa de ingeniería de características y pre-procesamiento, se procedió con la fase de implementación de los modelos de aprendizaje. De acuerdo al marco teorico establecido, se evaluaron cinco modelos distintos: Máquinas de Soporte Vectorial (SVM), Árboles de Decisión, Bolsa de Árboles, Bosques Aleatorios y la red neuronal recurrente de Memoria a Corto y Largo Plazo (LSTM). Con el propósito de garantizar una comparación objetiva y rigurosa en ambas estrategias de segmentación previamente descritas (Temporal y Aleatoria), se mantuvieron las mismas configuraciones de hiperparámetros para cada algoritmo, de esta manera aseguramos que las variaciones en el desempeño sean atribuidas exclusivamente a la naturaleza de la partición de datos y no a discrepancias en los ajustes de los modelos.

Adicionalmente, con el objetivo de evaluar el impacto de la reducción de dimensionalidad, la información fue sometida a una transformación mediante los métodos de PCA y PLS para posteriormente probar estas transformaciones en los modelos —a excepción de la red LSTM—. Los métodos de aplicación se detallarán en las secciones posteriores.

Por otro lado, para garantizar que cada algoritmo se ejecute en su punto de máxima eficiencia, la elección de la configuración óptima se realizó mediante la técnica de Grid Search[4] -a excepción de la red

[4] Scikit-learn: Gridsearchcv. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html, 2011a. Accedido el: 8 de marzo de 2026

LSTM- ya que este realiza una búsqueda exhaustiva de valores de parámetros específicos para un estimador.

4.7.1 *Reducción de dimensionalidad: PCA y PLS.*

Análisis de Componentes Principales Para la implementación de la transformación mediante PCA utilizamos un umbral de varianza explicada acumulada del 90 %. Bajo este criterio, nuestro algoritmo identificó que el número mínimo de eigenvalores necesarios para preservar la estructura esencial de nuestra información original es de tan solo dos componentes principales (Figura 4.6) frente a los cinco que originalmente establecimos. Esta simplificación nos puede ayudar a agilizar el proceso de optimización y convergencia en los modelos de aprendizaje.

Mínimos Cuadrados Parciales Con la transformación de PLS se determinó extraer tres componentes latentes, los cuales permiten que los modelos de aprendizaje trabajen en un espacio reducido de solo tres dimensiones, asegurando que en ellos se contenga una relevancia estadística probada respecto a las variaciones de la variable dependiente, lo cual también nos apoya en el proceso de optimización de los algoritmos de machine learning.

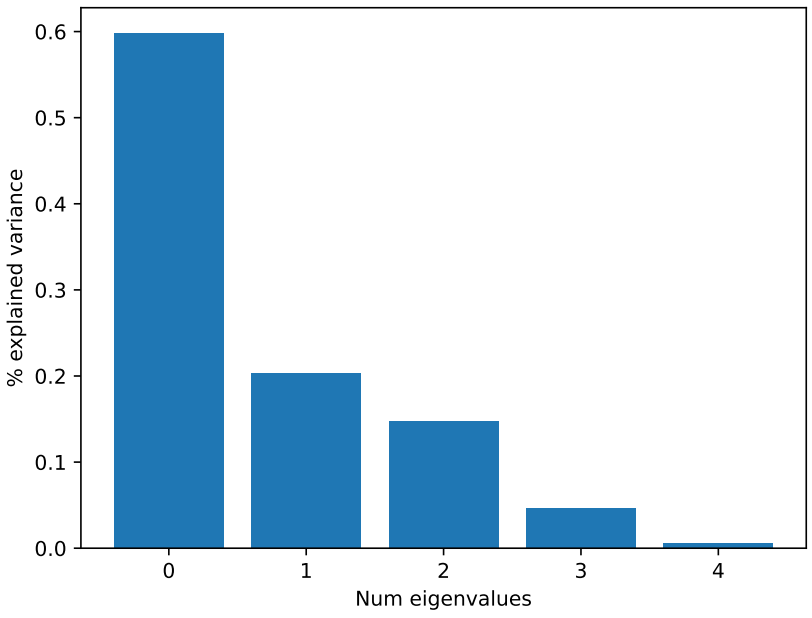

Figura 4.6: *Análisis de Componentes Principales (PCA): total de eigenvalores*

Modelo	Hiperparámetros y Rangos de Búsqueda
SVR	Kernel: {linear, poly, rbf, sigmoid}; C: {0.01, 0.1, 1, 10, 100, 1000}; Gamma: {scale, auto}; Degree: {2, 3, 4, 5, 6}; Epsilon: {0.01, 0.1, 0.2}.
Árbol de Decisión	Splitter: {best}; Max_depth: {1-15}; Min_samples_split: {2, 5, 10, 15, 20, 25, 30}; Min_samples_leaf: {1-14}; CCP_alpha: {0.0, 0.01, 0.1}; Criterion: {squared_error, friedman_mse}.
Bolsa de Árboles	N_estimators: {10, 50, 100, 200}; Max_samples: {0.6, 0.7, 0.8}; Max_depth: {4, 6, 10}; Min_samples_split: {15, 20, 30}; Min_samples_leaf: {4, 6, 10, 15}.
Bosque Aleatorio	N_estimators: {100, 200, 300, 400, 500}; Max_depth: {4, 6, 8, 10}; Min_samples_split: {2-7}; Min_samples_leaf: {1-9}; Max_features: {auto, sqrt, log2}; Bootstrap: {True, False}.

Cuadro 4.7: Espacio de búsqueda (*Grid Search*) detallado para los modelos implementados.

4.7.2 *Ajuste de Hiperparámetros.*

La optimización de los modelos se basó en una búsqueda exhaustiva que abarcó los hiperparámetros que se detallaron en el capítulo 3. Como se puede observar en la Tabla 4.7, el espacio de búsqueda se diseñó para cubrir un amplio aspecto de complejidad y de esta manera permitir identificar un equilibrio óptimo para lograr obtener los mejores resultados.

A diferencia de los modelos previos, la arquitectura para la Red Neuronal LSTM no fue sometida al proceso de Grid Search debido al elevado costo computacional que implica el ajuste de hiperparámetros. En su lugar, se implementó una estrategia de optimización heurística mediante la evaluación de 20 configuraciones estructurales distintas. El ajuste de estas redes se realizó de manera iterativa, basandose en el análisis de las curvas de aprendizaje y en la precisión de las métricas de desempeño que se detallarán en el siguiente capítulo. Las combinaciones de parámetros se muestran a continuación:

Cuadro 4.8: Bitácora detallada de las 20 configuraciones experimentales para la red LSTM.

Prueba	Neuronas	Épocas	LR	Batch	Optimizador
1	30	5,000	0.0001	32	Adam
2	30	5,000	0.0002	32	Adam
3	30	5,000	0.0005	32	Adam
4	60	5,000	0.0001	32	Adam
5	30	8,000	0.0001	32	SGD
6	50	8,000	0.0006	32	SGD
7	50	8,000	0.0009	32	SGD
8	30	8,000	0.0001548	32	SGD
9	70	8,000	0.0009	32	SGD
10	70	8,000	0.000954	32	SGD
11	50	8,000	0.0009	32	RMSprop
12	70	8,000	0.0005	32	RMSprop
13	70	8,000	0.0010	28	SGD
14	30	8,000	0.00009	32	SGD
15	30	1,000	0.0001	32	SGD
16	30	8,000	0.0001	50	SGD
17	50	8,000	0.0002	32	Adam
18	50	1,000	0.0009	32	RMSprop
19	50	8,000	0.0001	32	RMSprop
20	50	5,000	0.9999	32	RMSprop
22	50	8,000	0.0001	32	RMSprop

Después de haber concluido las etapas anteriormente descritas, se dio inicio a la fase de entrenamiento y validación de los modelos. La aplicación de métodos de reducción de dimensionalidad como PCA y PLS nos ayudó a optimizar el proceso de entrenamiento y...

El tiempo de ejecución de cada modelo varió dependiendo de la complejidad del algoritmo y del tamaño del conjunto de datos. En general, los modelos basados en árboles (Árbol de Decisión, Bolsa de Árboles y Bosque Aleatorio) tuvieron tiempos de entrenamiento más cortos en comparación con la red LSTM, la cual requirió un tiempo significativamente mayor debido a su arquitectura y al proceso iterativo de ajuste de hiperparámetros.

En el siguiente capítulo se presentan los resultados obtenidos para cada modelo bajo las dos estrategias de segmentación, así como un análisis comparativo de su desempeño utilizando métricas de evaluación como MSE y R^2. Además, se discuten las implicaciones de estos resultados en el contexto del nowcasting para la industria de los semiconductores.

5 Resultados y discusión

Contenido

Como se mencionó en el capítulo anterior, para hacer la evaluación de nuestros modelos se optó por considerar dos tipos de segmentación de nuestros datos: temporal y aleatoria. Para cada una de estas, se muestran a continuación los resultados obtenidos en los modelos implementados.

5.1 Resultados de la segmentación temporal

5.1.1 Máquinas de soporte vectorial

El modelo de Máquinas de Soporte Vectorial (SVR) presentó el desempeño más bajo en comparación con las demás arquitecturas evaluadas. La implementación de técnicas de reducción de dimensionalidad, como **PCA** o **PLS**, no logró mitigar la deficiencia

en la capacidad predictiva del algoritmo, resultando incluso en una méticas menos satisfactorias.

Como se detalla en la Tabla 5.1, el mejor resultado relativo se obtuvo con la configuración original, alcanzando un R^2 de **0.91** en el conjunto de entrenamiento. Sin embargo, el desempeño en los conjuntos de prueba y validación cruzada fue notablemente deficiente, con valores de R^2 negativos y MSE elevados. Este comportamiento confirma un caso crítico de **sobreajuste** y lo podemos ver gráficamente en la figura 5.1.

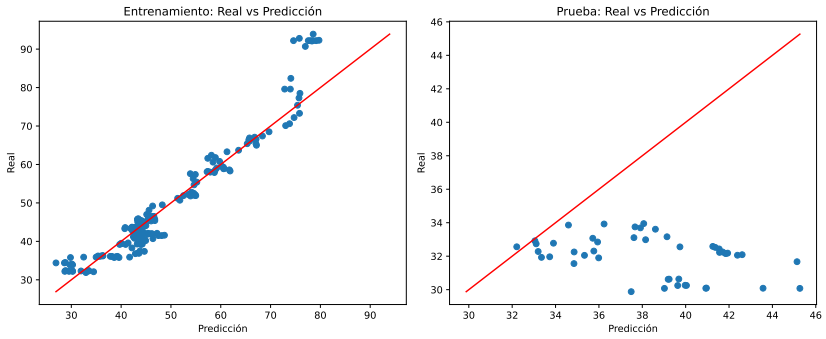

Figura 5.1: Ajuste de la recta a los datos de entrenamiento y prueba para el modelo de Máquinas de Soporte Vectorial con segmentación temporal.

La configuración óptima para este modelo se definió mediante un **Kernel RBF**, con un parámetro de regularización **C = 100**, un coeficiente **Gamma** bajo el criterio **Scale**, un **Epsilon** de **0.2** y un **Degree** de **2**.

Método	R^2 **Train**	**MSE Train**	R^2 **Test**	**MSE Test**	R^2 **CV**	**MSE CV**
Original	0.91	19.69	-39.24	57.09	-1122.12	31.60
PCA	0.89	23.27	-127.89	182.88	-7093.80	199.64
PLS	0.90	21.13	-62.84	90.58	-3147.11	88.58

Cuadro 5.1: Resultados del desempeño del modelo Máquinas de soporte vectorial con segmentación temporal.

5.1.2 *Árboles de decisión*

Con este modelo tampoco se alcanzaron niveles de desempeño satisfactorios, y el problema de sobreajuste sigue existiendo. En la tabla 5.2 podemos notar cómo nuestros valores de MSE y R^2 siguen siendo negativos en nuestros conjuntos de prueba y validación cruzada.

No obstante, se identificó que la transformación de datos mediante PLS ofrece una mejora en comparación con los datos originales y PCA. Específicamente, el error cuadrático medio sobre el conjunto de prueba se redujo de 5.22 a 3.65. Aunque esta disminución es estadísticamente favorable, el error sigue siendo elevado.

Para este experimento, los hiperparámetros óptimos seleccionados

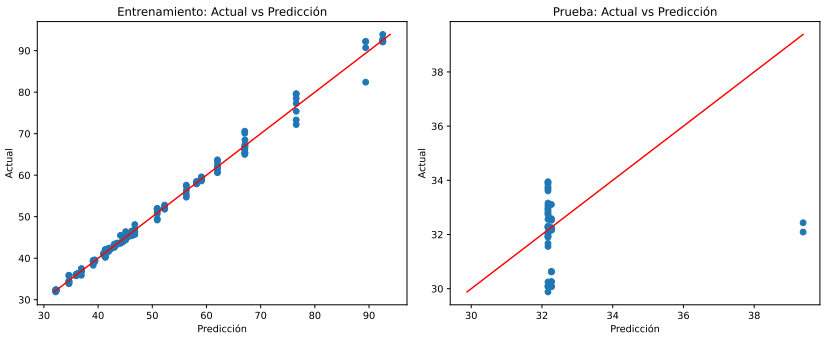

Figura 5.2: Ajuste de la recta a los datos de entrenamiento y prueba para el modelo de Árboles de Decisión con segmentación temporal.

definieron un valor de **alpha** de **0.0** y el uso de **squared_error** como criterio de división. Asimismo, se estableció una **profundidad máxima** de **15**, un **número mínimo de muestras por hoja** de **4** y un **mínimo de muestras para realizar una partición** de **10**, utilizando la estrategia **best** para el parámetro **splitter**. La complejidad derivada de estos parámetros, especialmente la profundidad del árbol, explica en gran medida la incapacidad del modelo para estabilizar sus predicciones en el horizonte de prueba.

El ajuste visual de estas predicciones frente a los valores reales se puede apreciar en la Figura 5.2.

Cuadro 5.2: Resultados del desempeño del modelo Árboles de decisión con segmentación temporal.

Método	R^2 **Train**	**MSE Train**	R^2 **Test**	**MSE Test**	R^2 **CV**	**MSE CV**
Original	0.99	0.64	-2.68	5.22	-713.98	20.11
PCA	0.98	3.69	-122.13	174.72	-213.71	6.04
PLS	0.99	1.12	-1.57	3.65	-216.39	6.11

5.1.3 *Bolsa de Árboles*

Bajo el esquema de división temporal, el modelo de Bolsa de Árboles revela un comportamiento de sobreajuste.

Como se observa en la Figura 5.3, el modelo se ajusta casi perfectamente a los datos de entrenamiento, pero falla al generalizar a los datos de prueba, lo que se refleja en un ajuste visual deficiente y métricas de desempeño negativas.

De acuerdo con los resultados presentados en la Tabla 5.3, aunque el modelo original alcanza un R^2 de 0.97 en el entrenamiento, el uso de la técnica de reducción de dimensionalidad PLS demostró ser la estrategia más efectiva para mitigar el error fuera de la muestra, reduciendo el MSE Test de 11.48 a 5.58.

La arquitectura óptima para este modelo se estableció con una

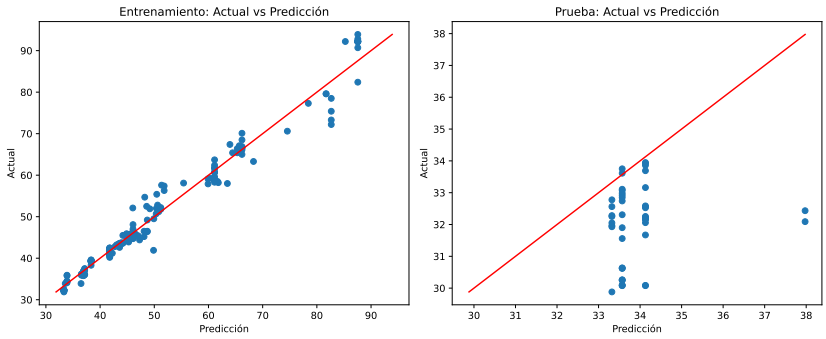

Figura 5.3: Ajuste de la recta a los datos de entrenamiento y prueba para el modelo de Bolsa de Árboles con segmentación temporal.

Método	R^2 Train	MSE Train	R^2 Test	MSE Test	R^2 CV	MSE CV
Original	0.97	4.76	-7.09	11.48	-747.78	21.07
PCA	0.93	15.46	-129.48	185.14	-10168.67	286.17
PLS	0.97	6.21	-2.93	5.58	-461.49	13.01

Cuadro 5.3: Resultados del desempeño del modelo Bolsa de Árboles con segmentación temporal.

profundidad máxima de **4** y un **mínimo de muestras por hoja** de **6**. Asimismo, se requirió un mínimo de **15** muestras para la **división interna**. El modelo se consolidó con **10 estimadores**, donde cada árbol fue entrenado utilizando un **subconjunto aleatorio** correspondiente al **70 %** de los datos originales, permitiendo así una mayor diversidad en el aprendizaje y una reducción en la varianza global del sistema.

5.1.4 *Bosques Aleatorios*

Con este modelo, se ve un ajuste casi perfecto durante la fase de entrenamiento con un R^2 de 0.99 y un ajuste extremadamente bueno a la recta como se muestra en la Figura 5.4.

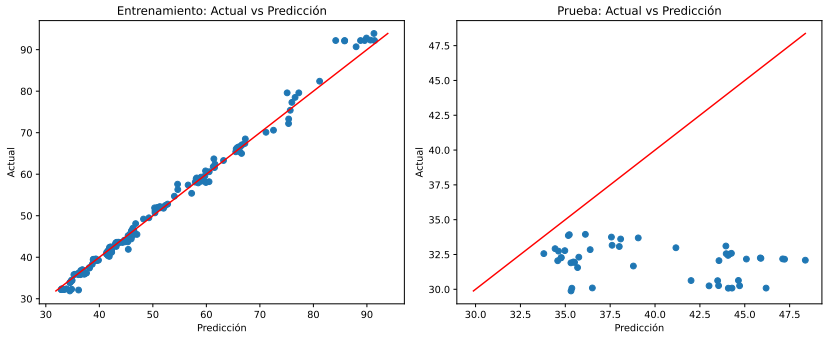

Figura 5.4: Ajuste de la recta a los datos de entrenamiento y prueba para el modelo de Bosques Aleatorios con segmentación temporal.

Un hallazgo importante es que, de acuerdo a la Tabla 5.4, el MSE obtenido en nuestra validación cruzada para la variante PLS es de

65.76, lo que representa una mejora significativa en comparación con el modelo original, que alcanzó un MSE de **110.22**. Este resultado sugiere que la aplicación de PLS contribuyó a reducir la varianza del modelo y a mejorar su capacidad de generalización, aunque en sus métricas de prueba haya sido mejor el modelo original.

Cuadro 5.4: Resultados del desempeño del modelo Bosques Aleatorios con segmentación temporal.

Método	R^2 **Train**	**MSE Train**	R^2 **Test**	**MSE Test**	R^2 **CV**	**MSE CV**
Original	0.99	0.56	-40.80	59.31	-3915.96	110.22
PCA	0.99	0.58	-145.83	208.34	-8877.40	249.83
PLS	0.99	2.06	-61.66	88.90	-2336.00	65.76

La optimización mediante Grid Search permitió determinar una configuración equilibrada que prioriza el control de la varianza. Se habilitó la técnica de **bootstrap** para el muestreo aleatorio con reemplazo, consolidando el ensamble con **100 estimadores**. Para mitigar el riesgo de sobreajuste sin perder profundidad analítica, se fijó una profundidad máxima de **8** y se utilizó el criterio **log2** para la selección de características en cada nodo. Finalmente, el modelo se ajustó con un mínimo de **una muestra por hoja** y **dos muestras para cada división**, maximizando la capacidad del bosque para capturar patrones complejos en los precios tecnológicos.

5.1.5 *Red Neuronal LSTM*

La red neuronal no fue la excepción al seguir la tendencia de sobreajuste que se ha observado en los modelos anteriores, sin embargo, los coeficientes de prueba mejoraron un poco. En la tabla 5.5 se pueden observar los resultados obtenidos en cada una de las 21 pruebas realizadas. Se puede concluir que el mejor modelo con respecto a las métricas de entrenamiento y prueba es la red neuronal número 11, a pesar de que el R^2 en entrenamiento no es el más alto, el desempeño en el conjunto de prueba es el mejor entre todas las pruebas realizadas, con un R^2 de **0.24** y un MSE de **1.06**. Y el error en validación cruzada es el más bajo entre todas las pruebas, con un MSE de **0.24**, aunque el R^2 sigue siendo negativo.

Con la figura 5.5 se puede observar el ajuste de la recta a los datos de entrenamiento y prueba, donde confirmamos de manera visual el fenómeno de sobreajuste, lo cual se refleja en las métricas obtenidas. Asimismo, la evaluación del error cuadrático medio durante el entrenamiento, como se muestra en la figura 5.6, revela una tendencia decreciente, lo que indica que el modelo está aprendiendo a ajustarse a los datos de entrenamiento y que nuestra tasa de aprendizaje fue adecuada ya que no se observa un comportamiento aleatorio o inestable,

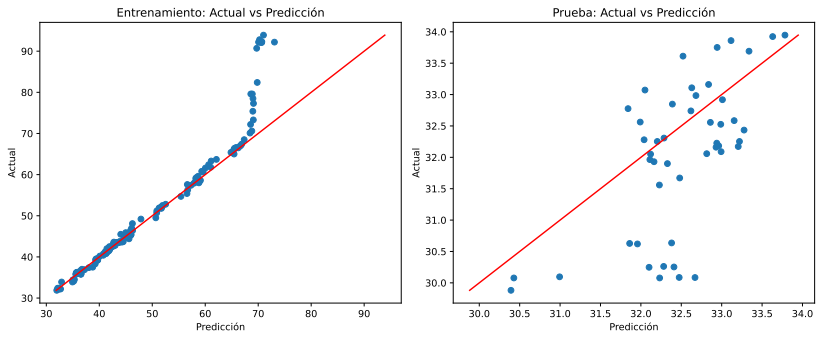

Figura 5.5: Ajuste de la recta a los datos de entrenamiento y prueba para la red Neuronal LSTM con mejor rendimiento con una segmentación temporal.

lo cual sugiere que el modelo pudo converger de manera eficiente. De hecho, se puede notar que a lo largo de las épocas el error en los datos de prueba siempre fue menor en comparación a los datos de entrenamiento.

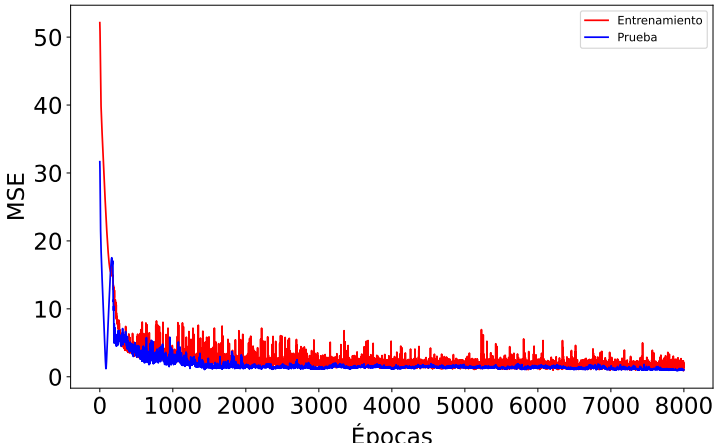

Figura 5.6: Evolución del error cuadrático medio durante el entrenamiento de la red Neuronal LSTM con segmentación temporal.

Cuadro 5.5: Resultados del desempeño de la red neuronal LSTM con segmentación temporal.

Prueba	R^2 **Train**	**MSE Train**	R^2 **Test**	**MSE Test**	R^2 **CV**	**MSE CV**
1	0.94	13.42	-8.53	13.53	-215.47	6.09
2	0.97	5.65	-18.04	27.02	-2434.36	68.53
3	0.94	13.77	-20.59	30.63	-2064.88	58.13
4	0.97	5.66	-7.34	11.83	-472.38	13.32
5	0.94	11.91	-16.47	24.79	-1782.62	50.19

Continuación de la Tabla 5.5

Prueba	R^2 Train	MSE Train	R^2 Test	MSE Test	R^2 CV	MSE CV
6	0.86	30.59	-2.93	5.57	-91.63	2.60
7	0.86	30.96	-14.01	21.29	-1410.11	39.70
8	0.96	8.60	-37.54	54.68	-3637.38	102.38
9	0.84	35.47	-17.62	26.42	-2957.95	83.26
10	0.85	33.66	-1.16	3.07	-79.67	2.27
11	0.87	29.18	0.24	1.06	-7.54	0.24
12	0.86	30.21	-1.07	2.95	-535.13	15.08
13	0.85	33.60	-2.87	5.49	-1145.93	32.27
14	0.97	5.83	-8.13	12.96	-219.40	6.20
15	0.94	12.70	-16.12	24.30	-588.84	16.59
16	0.97	6.80	-10.29	16.03	-156.95	4.44
17	0.88	25.63	-2.54	5.02	-355.77	10.03
18	0.93	16.15	-0.32	1.88	-66.42	1.89
19	0.91	18.74	-1.70	3.84	-257.94	7.28
20	0.61	90.03	-22.66	33.57	-6701.90	188.61
21	0.90	21.73	-3.48	6.36	-276.81	7.81

5.2 *Resultados de la segmentación aleatoria*

5.2.1 *Máquinas de soporte vectorial*

A diferencia de la segmentación temporal, el modelo de Máquinas de Soporte Vectorial con segmentación aleatoria mostró una mejora significativa en las métricas de prueba, aunque el problema no se solucionó del todo. En la tabla 5.6 se puede observar que el modelo original alcanzó un R^2 de **0.98** en el conjunto de prueba, lo que representa una mejora sustancial en comparación con la segmentación temporal. Sin embargo, el desempeño en validación cruzada sigue siendo deficiente, con un R^2 negativo y un MSE elevado, lo que indica que el modelo aún no generaliza bien a datos no vistos.

Método	R^2 Train	MSE Train	R^2 Test	MSE Test	R^2 CV	MSE CV
Original	0.99	0.83	0.98	2.63	-470.52	13.26
PCA	0.90	24.26	0.91	18.74	-1284.16	36.16
PLS	0.99	2.21	0.97	4.68	-492.35	13.88

Cuadro 5.6: Resultados del desempeño del modelo Máquina de Soporte Vectorial con segmentación aleatoria.

Para este experimento, las mejores métricas se obtuvieron con una configuración de **Kernel RBF**, un parámetro de regularización **C = 100**,

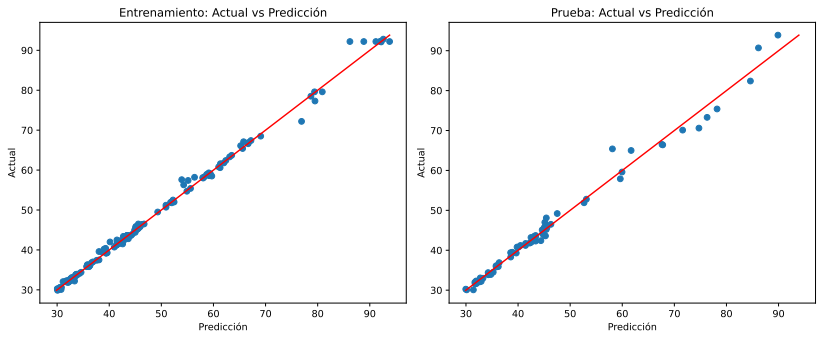

Figura 5.7: Ajuste de la recta a los datos de entrenamiento y prueba para el modelo de Máquinas de Soporte Vectorial con segmentación aleatoria.

un coeficiente **Gamma** bajo el criterio **Scale**, un **Epsilon** de **0.2** y un **Degree** de **2**, usando los datos sin ninguna transformación.

5.2.2 *Árboles de decisión*

Para este modelo, se observó que los datos pudieron ajustarse de manera perfecta en el conjunto de entrenamiento, alcanzando un R^2 de **1.00** y un MSE de **0.00**, utilizando el conjunto de datos original. Para el conjunto de prueba, no se tuvo un mal desempeño, pero sí se vio una baja, ya que el MSE se incrementó a **8.02** y el R^2 disminuyó a **0.96**. Sin embargo, el desempeño de validación cruzada dejó mucho que desear, con un R^2 negativo de **-29.34** y un MSE de **0.85**, lo que confirma la presencia de sobreajuste en el modelo y la incapacidad del modelo de explicar la variabilidad de los datos fuera de la muestra.

Cuadro 5.7: Resultados del desempeño del modelo de Árbol de Decisión con segmentación aleatoria.

Método	R^2 **Train**	**MSE Train**	R^2 **Test**	**MSE Test**	R^2 **CV**	**MSE CV**
Original	1.00	0.00	0.96	8.02	-29.34	0.85
PCA	0.91	20.33	0.80	39.90	-11.33	0.34
PLS	0.99	0.15	0.93	14.66	-11.33	0.34

Como se mencionó, el mejor modelo fue el que se entrenó con los datos originales, utilizando una configuración de hiperparámetros que definió un valor de **alpha** de **0.0** y el uso de **friedman_mse** como criterio de división. Asimismo, se estableció una **profundidad máxima** de **15**, un **número mínimo de muestras por hoja** de **1** y un **mínimo de muestras para realizar una partición** de **2**, utilizando la estrategia **best** para el parámetro **splitter**. La complejidad derivada de estos parámetros, especialmente la profundidad del árbol, explica en gran medida la incapacidad del modelo para estabilizar sus predicciones en el horizonte de prueba.

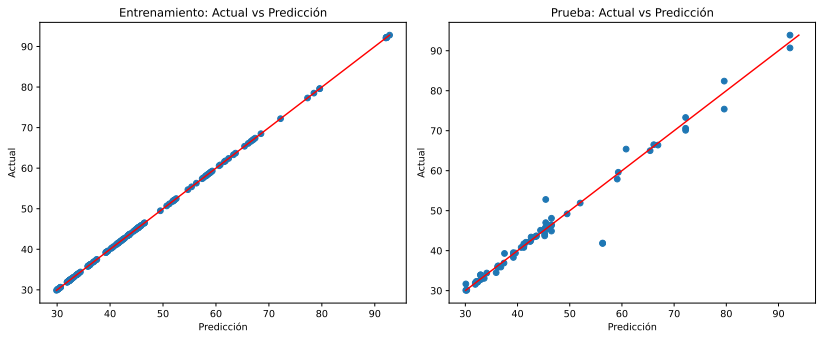

Figura 5.8: Ajuste de la recta a los datos de entrenamiento y prueba para el modelo de Árboles de Decisión con segmentación aleatoria.

5.2.3 *Bolsa de Árboles*

Este modelo mostró un desempeño similar al de los árboles de decisión, con un ajuste casi perfecto a los datos de entrenamiento, alcanzando un R^2 de **0.98** y un MSE de **4.52**, usando los datos sin ningún tipo de transformación. Sin embargo, el desempeño en el conjunto de prueba bajó y se obtuvo un R^2 de **0.93** y un MSE de **13.26**. Además, la validación cruzada reveló un desempeño deficiente, con un R^2 negativo de **-32.99** y un MSE de **0.95**, lo que confirma la presencia de la tendencia de sobreajuste en los datos.

Método	R^2 **Train**	**MSE Train**	R^2 **Test**	**MSE Test**	R^2 **CV**	**MSE CV**
Original	0.98	4.52	0.93	13.26	-32.99	0.95
PCA	0.88	28.23	0.74	52.92	-69.27	1.97
PLS	0.95	11.50	0.87	26.07	-34.11	0.98

Cuadro 5.8: Resultados del desempeño del modelo Bolsa de Árboles con segmentación aleatoria.

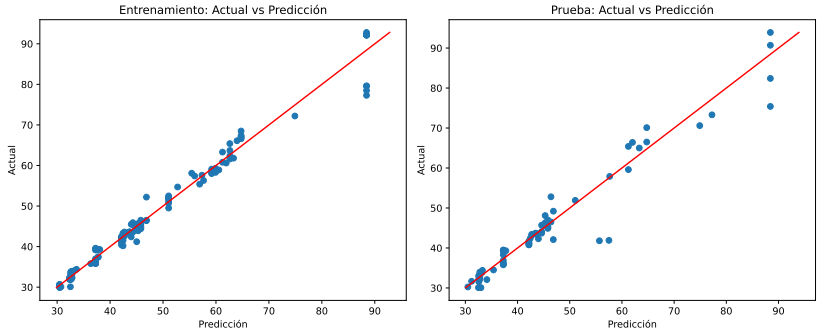

Figura 5.9: Ajuste de la recta a los datos de entrenamiento y prueba para el modelo de Bolsa de Árboles con segmentación aleatoria.

El modelo se consolidó con una configuración de hiperparámetros que estableció una **profundidad máxima** de **4** y un **mínimo de muestras por hoja** de **4**. Asimismo, se requirió un mínimo de **15** muestras para la

división interna. El modelo se utilizó **10 estimadores**, donde cada árbol fue entrenado utilizando un **subconjunto aleatorio** correspondiente al **80 %** de los datos originales, permitiendo así una mayor diversidad en el aprendizaje y una reducción en la varianza global del sistema.

5.2.4 *Bosques Aleatorios*

El modelo de Bosques Aleatorios fue de los pocos modelos que mostró buenos resultados con la transformación de datos con PCA y PLS, ya que las tres pruebas realizadas lograron un R^2 de **0.99**, un resultado casi perfecto, aunque en el conjunto de prueba el modelo original fue el que obtuvo un mejor desempeño, con un R^2 de **0.94** y un MSE de **11.08**. Pero el sobreajuste siguió siendo un problema en la validación cruzada.

Método	R^2 **Train**	**MSE Train**	R^2 **Test**	**MSE Test**	R^2 **CV**	**MSE CV**
Original	0.99	0.00	0.94	11.08	-81.45	2.32
PCA	0.99	2.19	0.75	52.26	-2155.27	60.67
PLS	0.99	0.66	0.91	17.82	-1211.71	34.12

Cuadro 5.9: Resultados del desempeño del modelo Bosques Aleatorios con segmentación aleatoria.

La optimización mediante Grid Search utilizó la técnica de **bootstrap** para el muestreo aleatorio con reemplazo, consolidando el ensamble con **100 estimadores**. Para mitigar el riesgo de sobreajuste sin perder profundidad analítica, se fijó una profundidad máxima de **10** y se utilizó el criterio **log2** para la selección de características en cada nodo. Finalmente, el modelo se ajustó con un mínimo de **una muestra por hoja** y **tres muestras para cada división**.

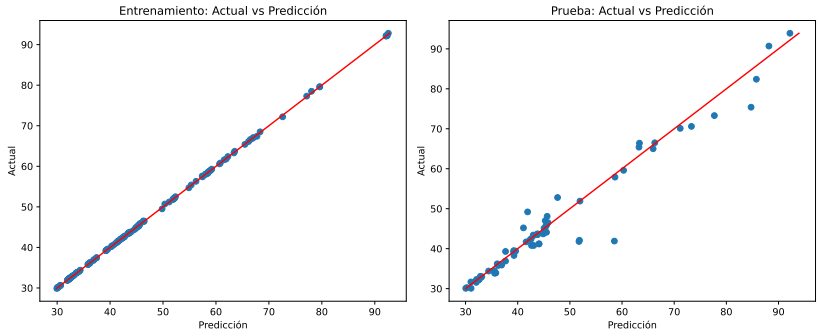

Figura 5.10: Ajuste de la recta a los datos de entrenamiento y prueba para el modelo de Bosques Aleatorios con segmentación aleatoria.

5.2.5 *Red Neuronal LSTM*

Finalmente, la arquitectura **LSTM** presentó resultados de especial interés. Si bien el sobreajuste persistió como un desafío inherente a

la complejidad del modelo, se observó una alta variabilidad en las métricas de desempeño. En el conjunto de entrenamiento, nuestra R^2 osciló entre un valor máximo de **0.98** y un mínimo de **0.41**. Por otra parte, en el conjunto de prueba, los valores fluctuaron entre **0.97** y **0.22**. Estas cifras sugieren que, bajo configuraciones específicas, las redes neuronales recurrentes poseen la capacidad de capturar la dinámica subyacente de los precios del IPP.

No obstante, el análisis de validación cruzada reveló nuevamente limitaciones en la estabilidad general, con valores de R^2 negativos en la mayoría de las iteraciones. A pesar de esta tendencia, la configuración correspondiente a la **Prueba 5** destacó significativamente sobre las demás.

Cuadro 5.10: Resultados del desempeño de la arquitectura LSTM con segmentación aleatoria.

Prueba	R^2 Train	MSE Train	R^2 Test	MSE Test	R^2 CV	MSE CV
1	0.84	38.24	0.87	26.94	-12.08	0.36
2	0.97	6.11	0.95	10.09	-10.03	0.31
3	0.98	3.75	0.97	5.21	-12.50	0.38
4	0.97	5.23	0.95	9.17	-14.46	0.43
5	0.98	4.24	0.97	6.09	0.45	0.01
6	0.90	22.56	0.91	17.16	-10.70	0.32
7	0.93	16.18	0.94	11.86	-60.75	1.73
8	0.98	4.13	0.96	7.24	-43.98	1.26
9	0.89	26.32	0.90	20.24	-35.14	1.01
10	0.88	27.50	0.90	20.93	-49.15	1.41
11	0.89	25.59	0.91	18.18	-13.05	0.39
12	0.89	25.56	0.91	18.49	-17.81	0.52
13	0.90	23.16	0.91	18.16	-32.97	0.95
14	0.98	4.42	0.96	6.80	-0.01	0.02
15	0.95	10.52	0.95	10.48	-22.71	0.66
16	0.98	3.33	0.96	6.72	-9.36	0.29
17	0.97	7.39	0.96	7.96	-8.34	0.26
18	0.97	5.27	0.96	6.47	-5.46	0.18
19	0.97	5.58	0.96	6.45	-7.01	0.22
20	0.41	145.78	0.22	162.29	-1282.25	36.11
21	0.97	6.45	0.96	6.60	-12.10	0.36

La red neuronal de la **Prueba 5** demostró un desempeño sobresaliente, alcanzando un R^2 de **0.98** en entrenamiento y **0.97** en

prueba. Resulta particularmente notable su error cuadrático medio (**MSE**) en validación cruzada de apenas **0.01**. Aunque la capacidad de generalización en CV sigue siendo un área de oportunidad, con un R^2 de **0.45**, esta configuración se posiciona como la más robusta de todo el estudio para la predicción de los precios del IPP.

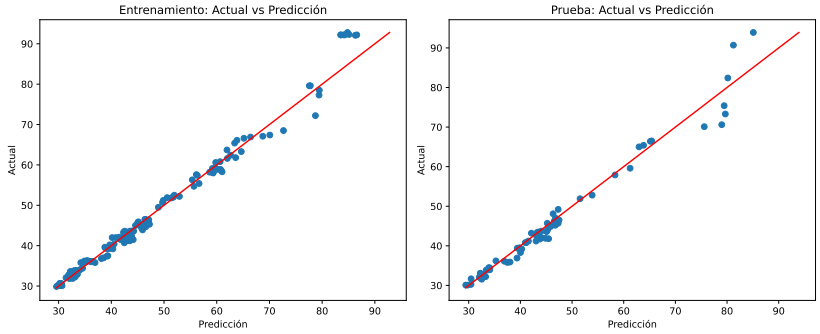

Figura 5.11: Ajuste de la recta a los datos de entrenamiento y prueba para el modelo de Red Neuronal LSTM con segmentación aleatoria.

Como se ilustra en la Figura 5.11, el ajuste de la recta entre los valores reales y las predicciones para la red número 5 confirma visualmente la estrecha relación encontrada por el modelo. De manera complementaria, la evolución del MSE durante el entrenamiento (Figura 5.12) evidencia un proceso de aprendizaje convergente y estable. La ausencia de fluctuaciones erráticas indica un entrenamiento controlado, validando que el número de épocas seleccionado fue suficiente para alcanzar la convergencia; un incremento en este parámetro solo habría resultado en un costo computacional redundante sin beneficios marginales en la precisión.

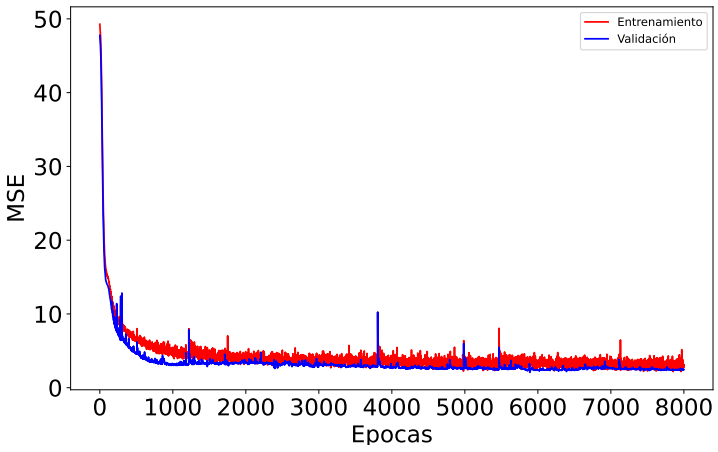

Figura 5.12: Evolución del error cuadrático medio durante el entrenamiento para el modelo de Red Neuronal LSTM con segmentación aleatoria.

5.3 *Análisis comparativo de modelos*

Tras evaluar las diversas arquitecturas de modelos de aprendizaje automático es evidente que los modelos tradicionales como son Máquinas de Soporte Vectorial, Árboles de Decisión, Bolsa de Árboles y Bosques Aleatorios muestran una tendencia propensa al sobreajuste, ya que se pudieron lograr precisiones casi perfectas en el conjunto de entrenamiento pero fallando de manera drástica en el conjunto de prueba y validación cruzada. Por otro lado la red neuronal de manera general demostró una capacidad mayor para capturar la dinámica de los datos, ya que este tipo de arquitectura está diseñada para capturar dependencias secuenciales y patrones temporales que sirven ampliamente para series financieras y económicas.

Un factor determinante en la interpretación de los resultados fue la segmentación de los datos. En la segmentación temporal, se observó una degradación en el desempeño de nuestras métricas debido al fenómeno llamado **Concept Drift**, que se refiere a un escenario donde la relación entre las variables de entrada y la variable objetivo cambia a través del tiempo, lo que provoca una disminución en la precisión de los modelos, ya que los patrones aprendidos dejan de ser válidos para los datos futuros[1]. En contraste, la segmentación aleatoria produjo métricas muy buenas y elevadas, pero podrían considerarse engañosas debido al **Data Leakage**, que ocurre cuando el modelo aprende de información futura o datos que no estarán disponibles en el momento de la predicción, lo que da como resultado resultados aparentemente buenos pero que no se replicarán en un entorno de producción[2].

Finalmente, es importante destacar que en las series temporales no lineales y ruidosas como la de los precios del IPP, la obtención de métricas como la del R^2 puede ser un desafío, de hecho tienden a no tener un valor demasiado alto y no necesariamente son un indicador de un mal modelo. Algunas literaturas [3] sugieren que, en datos con varianza muy alta y componentes ruidosos, el R^2 suele ser bajo debido a que gran parte de la fluctuación diaria es impredecible, lo que nos lleva a priorizar métricas como el MSE como una medida más adecuada para evaluar el desempeño de los modelos en este contexto.

Dicho lo anterior, el modelo que mostró un mejor desempeño al obtener un menor error en la validación cruzada fue sin duda la red LSTM número 5 usando la segmentación aleatoria y utilizando una configuración de hiperparámetros de **30 neuronas**, una **tasa de aprendizaje** de **0.0001**, un número de **épocas** de **8,000** y un **batch size** de **32**, utilizando los datos sin ninguna transformación y usando el **optimizador SGD**. Pero antes de concluir que este modelo es el mejor, es importante realizar un análisis con las pruebas de Rolling Walk-Forward, PBO y Bootstrap Reality check, las cuales describimos en

[1] João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Hamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, 46, 04 2014. DOI: 10.1145/2523813

[2] IBM. What is data leakage in machine learning?, 2024. URL https://www.ibm.com/think/topics/data-leakage-machine-learning. Accedido el 3 de abril de 2026

[3] Brett W. Gelino, Justin C. Strickland, and Matthew W. Johnson. R2 should not be used to describe behavioral-economic discounting and demand models. *Journal of the Experimental Analysis of Behavior*, 122(2):117–138, 2024. DOI: https://doi.org/10.1002/jeab.4200. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/jeab.4200

el capítulo 3.4. Para las siguientes pruebas, pondremos en acción al modelo que creemos que es el ganador simulando un escenario real.

5.3.1 *Prueba de Rolling Walk- Forward*

Debido a que el modelo ganador fue entrenado con una segmentación aleatoria, es importante evaluar su desempeño en un escenario más realista, es decir, con una serie temporal que simule un entorno de producción. Para esto, se implementó la prueba de Rolling Walk-Forward. Conforme a la fundamentación teórica de este método, nuestro conjunto de datos se dividió respetando estrictamente el orden temporal. El modelo fue evaluado sobre el set de test que actúa como el futuro relativo. En cada iteración, se incorporó un nuevo dato al contexto del modelo para predecir el valor del paso siguiente, permitiendo capturar la dinámica de cambio en la tendencia del IPP.

Debido al alto costo computacional que implica el re-entrenamiento de la red neuronal LSTM en cada paso (como se ilustra de manera conceptual en la Figura3.1) se optó por una estrategia de inferencia dinámica. Bajo este enfoque, el modelo no se re-entrenó en cada iteración; en su lugar, se utilizó su capacidad de memoria para evaluar el día siguiente empleando los datos precedentes como contexto secuencial.

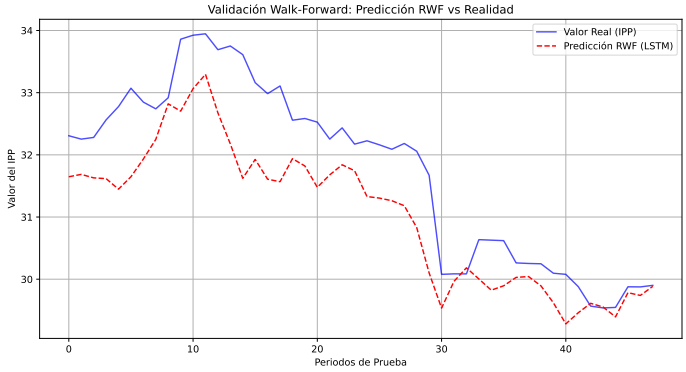

Figura 5.13: Validación Rolling Walk-Forward con la red neuronal LSTM configuración 5

Los resultados obtenidos en la prueba mostraron un Error Cuadrático Medio de **0.75** (Figura 5.14). Al comparar esta métrica con la obtenida en validaciones previas, se observa una representación más fiel de la capacidad del modelo. Como se observa en la Figura 5.13, la red presenta un ajuste progresivo hacia el final del horizonte de estudio, logrando capturar con mayor precisión los datos reales en los últimos seis meses, período que se mantuvo estrictamente fuera del proceso de entrenamiento y optimización de hiperparámetros.

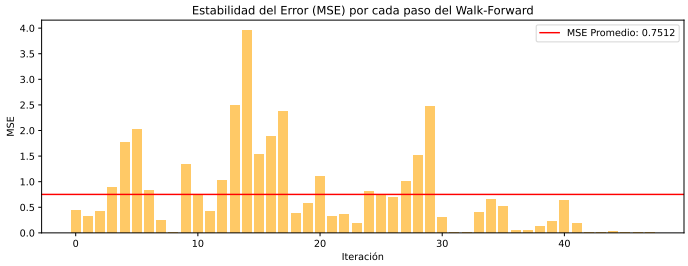

Figura 5.14: Visualización grafica del Error Cuadrático Medio en la validación Rolling Walk-Forward

5.3.2 *Prueba de Boostrap Reality check*

Para determinar si la capacidad predictiva de nuestra red ganadora es estadísticamente superior al azar o a modelos de referencia simplistas, se implementó la prueba de Bootstrap Reality Check. Siguiendo el modelo de parsimonia, se utilizó como benchmark un Modelo Naive de persistencia, como lo hacen los autores Beck et al. [2025], el cual asume que el mejor predictor para el tiempo t + 1 es simplemente el valor observado en el tiempo t.

Los resultados de la prueba arrojan un **diferencial de MSE de -0.63** y un **p-valor de 1.00**. Debido a que el p-valor es superior al umbral de significancia de 0.05, no existe evidencia estadística suficiente para rechazar la hipótesis nula. Esto implica que, en el horizonte de tiempo evaluado, el modelo LSTM no logra superar al modelo Naive.

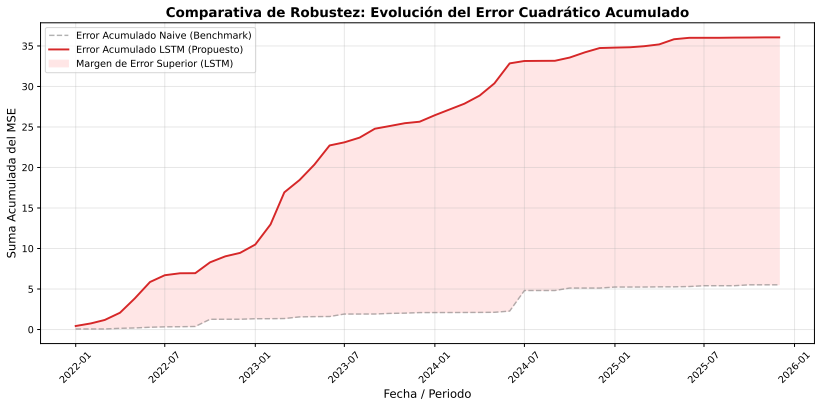

Figura 5.15: Error acumulado de la prueba Bootstrap Reality Check: Modelo Naive vs Red Neuronal LSTM.

Como se observa en la figura 5.15, la línea del modelo Naive se mantiene por debajo de la red LSTM. Este fenómeno, confirma que la naturaleza de la variable objetivo (IPP) representa un reto significativo para la aplicación de modelos de Machine Learning. Esta dificultad predictiva puede atribuirse a que las fluctuaciones diarias de las variables independientes pueden ser interpretadas como ruido en lugar

de señales predictivas útiles. En consecuencia, el modelo Naive de persistencia resulta ser un competidor robusto, ya que aprovecha la alta autocorrelación de la serie sin incurrir en los errores de sobreajuste de variaciones mínimas.

5.3.3 *Probabilidad de Sobreajuste en el Backtest (PBO)*

Para la prueba de sobreajuste, incluimos los mejores modelos de la segmentación aleatoria. Los resultados finales muestran un **PBO de 1.31 %**, un valor significativamente bajo que fortalece la validez de la metodología empleada y reduce el riesgo de sesgo por elección aleatoria.

Este resultado indica que existe una probabilidad mínima de que la selección del modelo óptimo sea producto de un ajuste espurio al ruido de los datos históricos. La distribución de los logits se encuentra marcadamente desplazada hacia la izquierda del umbral de sobreajuste, lo que demuestra que los modelos con mejor desempeño en el conjunto de entrenamiento mantienen su capacidad predictiva y ranking relativo en escenarios no observados (Out-of-Sample). Al situarse muy por debajo del límite convencional del 5 %, se confirma la robustez del protocolo de backtesting y se garantiza que los patrones identificados para el IPP de circuitos integrados poseen un valor estadístico real más allá de la coincidencia algorítmica.

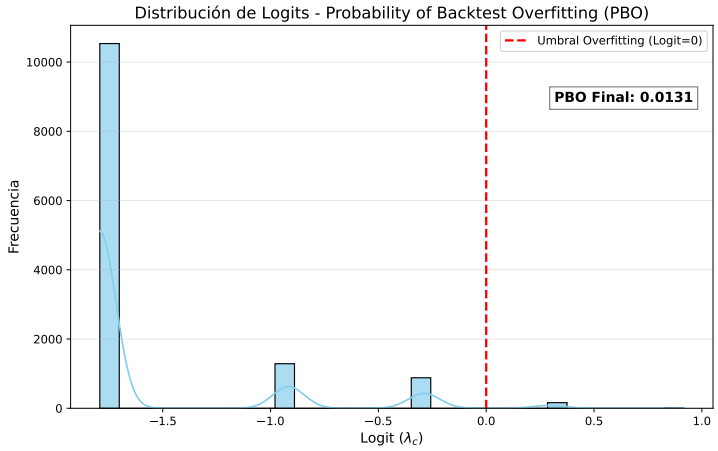

Figura 5.16: Se observa un sesgo negativo dominante en los logits, lo que representa una baja probabilidad (1.31 %) de que el rendimiento del modelo seleccionado sea resultado de un sobreajuste en el proceso de optimización.

6 Conclusiones y trabajo futuro

La presente investigación permitió evaluar la viabilidad de implementar modelos de Machine Learning para el nowcasting del Índice de Precios al Productor (IPP) en el sector electrónico, enfocándonos en los circuitos integrados. Si bien el uso de redes neuronales recurrentes como la LSTM representa una propuesta innovadora, y aparentemente ganadora de acuerdo a nuestras métricas de evaluación, la realidad es que la naturaleza econométrica de la variable objetivo impone desafíos significativos. El IPP, al ser un índice que se construye con diferentes estructuras de costos, contratos de largo plazo y variables geopolíticas, no siempre puede ser capturada de forma eficiente por modelos diseñados para detectar patrones.

Uno de los hallazgos más relevantes fue la discrepancia entre las frecuencias de las variables. Mientras que las variables independientes presentan una alta volatilidad diaria, el IPP manifiesta cambios más lentos. Esta asimetría genera una relación ruidosa desfavorable para los modelos seleccionados, donde la red neuronal tiende a sobreajustarse al ruido diario en lugar de identificar tendencias. Esto se confirmó en el capítulo anterior, donde las métricas de validación cruzada mostraron un desempeño inferior al esperado frente a datos no vistos. Además, es importante mencionar que la reducción de dimensionalidad no fue un factor determinante para la mejora del desempeño de ninguno de nuestros modelos, lo cual sugiere que la cantidad de variables no es la principal limitante, sino la calidad de la señal que estas variables pueden aportar.

Las pruebas de robustez aplicadas, aportaron una perspectiva realista y crítica sobre el desempeño del modelo ganador en un entorno de producción real. El hecho de que el modelo Naive de persistencia no fuera superado estadísticamente demuestra la importancia de la autocorrelación serial en el IPP.

En conclusión, el uso de modelos de Machine Learning de alta complejidad no se traduce en una mejora significativa de los resultados para la predicción del IPP y aplicación del nowcasting. Frente a este tipo de variables, modelos más simples o de carácter econométrico podrían ser una opción más recomendable, ya que logran capturar la

tendencia de fondo y gestionar el ruido de las variables independientes de manera más eficiente y con un menor costo computacional.

Dicho lo anterior y de acuerdo al cumplimiento de los objetivos de esta investigación, nuestro modelo ganador puede ser implementado como una herramienta de soporte a la toma de decisiones y no como un sistema de predicción absoluta. Dado que la prueba de Rolling Walk-Forward demostró que la red requiere de un período de ajuste para alinearse a las tendencias del IPP, su valor estratégico radica en su capacidad de servir como una guía dinámica.

Esto es muy valioso en la gestión de la cadena de suministro ya que el anticipar posibles fluctuaciones en los costos de materiales de alto valor y con alta demanda como son los circuitos integrados facilita las negociaciones con clientes y proveedores, permitiendo a las organizaciones tener una postura proactiva al saber las tendencias actuales del mercado dadas por factores indirectos que afecta al costo de sus productos. Al entender que el modelo aprende y se ajusta conforme avanza el tiempo, directivos o encargados de la toma de decisiones pueden utilizar estos resultados para establecer márgenes de seguridad en los precios para asegurar una rentabilidad frente a la volatilidad de los insumos.

En última instancia, la integración de modelos como el nowcasting en la planificación financiera ayuda a reducir la incertidumbre en el mercado. Aunque el modelo Naive sea un competidor fuerte en términos de error absoluto, el modelo de Machine Learning aporta una visión de direccionalidad, la cual es fundamental para la planificación y gestión de riesgos, permitiendo que la incertidumbre financiera se transforme en un riesgo medible y gestionable.

Bibliografía

Chinese yuan renminbi to u.s. dollar spot exchange rate. https://fred.stlouisfed.org/series/DEXCHUS.

Factset website. https://www.factset.com/.

Scikit-learn: Gridsearchcv. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html, 2011a. Accedido el: 8 de marzo de 2026.

Scikit-learn: Svr. https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html, 2011b. Accedido el: 16 de febrero de 2026.

Scikit-learn: Bagging regressor. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingRegressor.html, 2011c. Accedido el: 16 de febrero de 2026.

Scikit-learn: Decision tree regressor. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html, 2011d. Accedido el: 16 de febrero de 2026.

Scikit-learn: Random forest regressor. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html, 2011e. Accedido el: 16 de febrero de 2026.

SEMI. (2024). Electronic system design industry posts $5.1 billion in revenue in q3 2024, esd alliance reports. URL https://www.semi.org/en/semi-press-release/electronic-system-design-industry-posts-dollar-5.1-billion-in-revenue-in-q3-2024-esd-alliance-reports. Consultado el 6 de septiembre de 2025.

Fortune Business Insights. (2025). Consumer electronics market size, share, trends, growth, 2032. https://www.fortunebusinessinsights.com/consumer-electronics-market-104693, a. Consultado el 31 de agosto de 2025.

Fortune Business Insights. (2025). Semiconductor market size, share, trends, growth, 2032. https://www.fortunebusinessinsights.com/

semiconductor-market-102365, b. Consultado el 13 de septiembre de 2025.

The Business Research Company. (2025). Market report image. https://www.thebusinessresearchcompany.com/report/analog-integrated-circuit-ic-global-market-report, c. Consultado el 14 de septiembre de 2025.

Ajay Agrawal, Joshua Gans, and Avi Goldfarb. *The Economics of Artificial Intelligence: An Agenda*. University of Chicago Press, 2019. DOI: 10.7208/chicago/9780226613475.001.0001.

Zakaria Alameer, Mohamed Abd Elaziz, Ahmed A. Ewees, Haiwang Ye, and Zhang Jianhua. Forecasting gold price fluctuations using improved multilayer perceptron neural network and whale optimization algorithm. *Resources Policy*, 61:250–260, 2019. ISSN 0301-4207. DOI: https://doi.org/10.1016/j.resourpol.2019.02.014.

David H. Bailey, Jonathan M. Borwein, Marcos López de Prado, and Qiji Jim Zhu. The probability of backtest overfitting. *The Journal of Computational Finance*, 20(4):39–69, 2017. DOI: 10.21314/JCF.2016.322. URL https://doi.org/10.21314/JCF.2016.322.

Nico Beck, Jonas Dovern, and Stefanie Vogl. Mind the naive forecast! a rigorous evaluation of forecasting models for time series with low predictability. *Applied Intelligence*, 55, 02 2025. DOI: 10.1007/s10489-025-06268-w.

Mike Bernico. *Deep Learning Quick Reference: Useful hacks for training and optimizing deep neural networks with TensorFlow and Keras*. Packt Publishing Ltd., 2018.

K. Browning and Chris Collier. Nowcasting of precipitating systems. *Reviews of Geophysics - REV GEOPHYS*, 27, 01 1989. DOI: 10.1029/RG027i003p00345.

Felix Francisco Casares. Nowcasting: Modelos de factores dinámicos y ecuaciones puente para la proyección del pib del ecuador. Technical report, COMPENDIUM, ISSN Online 1390-9894, Volumen 4, N° 8, Agosto, 2017, pp 45 – 66, 2017.

M. E. Correal. Modelo factorial dinámico tar. Recuperado de Repositorio Institucional de la Universidad Nacional de Colombia, 2007. Accedido: 2025-09-19.

Laura D'Amato, Lorena Garegnani, and Emilio Blanco. Nowcasting de pib: Evaluando las condiciones cíclicas de la economía argentina. Technical Report 2015/69, Banco Central de la República Argentina (BCRA), Investigaciones Económicas (ie), Buenos Aires, 2015.

Oscar de Jesús Gálvez-Soriano. Nowcasting del pib de méxico usando modelos de factores y ecuaciones puente. Technical report, Banco de México, 2018.

João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Hamid Bouchachia. A survey on concept drift adaptation. *ACM Computing Surveys (CSUR)*, 46, 04 2014. DOI: 10.1145/2523813.

Brett W. Gelino, Justin C. Strickland, and Matthew W. Johnson. R2 should not be used to describe behavioral-economic discounting and demand models. *Journal of the Experimental Analysis of Behavior*, 122(2):117–138, 2024. DOI: https://doi.org/10.1002/jeab.4200. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/jeab.4200.

Eric Ghysels, Virmantas Kvedaras, and Vaidotas Zemlys-Balevičius. Mixed data sampling (midas) regression models. In Hrishikesh D. Vinod and C.R. Rao, editors, *Financial, Macro and Micro Econometrics Using R*, volume 42 of *Handbook of Statistics*, pages 117–153. Elsevier, 2020. DOI: 10.1016/bs.host.2019.01.005.

Lei Han, Juanzhen Sun, Wei Zhang, Yuanyuan Xiu, Hailei Feng, and Yinjing Lin. A machine learning nowcasting method based on real-time reanalysis data. *Journal of Geophysical Research: Atmospheres*, 122 (7):4038–4051, 2017. DOI: https://doi.org/10.1002/2016JD025783.

IBM. What is data leakage in machine learning?, 2024. URL https://www.ibm.com/think/topics/data-leakage-machine-learning. Accedido el 3 de abril de 2026.

Keras Team. Model training APIs. https://keras.io/api/models/model_training_apis/, 2023. Accedido el: 16 de febrero de 2026.

Max Kuhn and Kjell Johnson. *Applied Predictive Modeling*. Springer New York, NY, 1 edition, 2013. ISBN 978-1-4614-6848-6. DOI: 10.1007/978-1-4614-6849-3.

Jesus Lago, Grzegorz Marcjasz, Bart De Schutter, and Rafał Weron. Forecasting day-ahead electricity prices: A review of state-of-the-art algorithms, best practices and an open-access benchmark. *Applied Energy*, 293:116983, 2021. ISSN 0306-2619. DOI: https://doi.org/10.1016/j.apenergy.2021.116983.

Roberto S. Mariano and Yasutomo Murasawa. A new coincident index of business cycles based on monthly and quarterly series. *Journal of Applied Econometrics*, 18(4):427–443, None 2003. DOI: 10.1002/jae.695.

Brahami Menaouer, Fatma Zahra Abdeldjouad, Mohammed Sabri, Khalissa Semaoune, and Matta Nada. Forecasting supply chain

demand approach using knowledge management processes and supervised learning techniques. *International Journal of Information Systems and Supply Chain Management*, 15:1–21, 01 2022. DOI: 10.4018/IJISSCM.2022010103.

Aamirah Mohammed and Sardar Asif Khan. Global disruption of semiconductor supply chains during covid-19: An evaluation of leading causal factors. Volume 2: Manufacturing Processes; Manufacturing Systems:V002T06A011, 06 2022.

U.S. Bureau of Labor Statistics. URL https://www.bls.gov/ppi/.

Hayrettin Okut. *Deep Learning: Long-Short Term Memory*. 06 2021.

Szymon Piotrowski. Critical raw materials in the semiconductor industry: Current challenges and perspectives. pages 65–72, 2023. Dr. Szymon Piotrowski, Adam Mickiewicz University in Poznań.

Secretaría de Economía de México. (2019). Fabricación de componentes electrónicos (Data México). URL https://www.economia.gob.mx/datamexico/es/profile/industry/semiconductor-and-other-electronic-component-manufacturing. Consultado el 31 de agosto de 2025.

Xingjian SHI, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-kin Wong, and Wang-chun WOO. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

SiamQuant. Walk-forward analysis, 2024. URL https://www.siamquant.com/walk-forward-analysis/. Accedido: 4 de abril de 2026.

Payal Soni, Yogya Tewari, and Deepa Krishnan. Machine learning approaches in stock price prediction: A systematic review. *Journal of Physics: Conference Series*, 2161(1):012065, jan 2022. DOI: 10.1088/1742-6596/2161/1/012065. URL https://doi.org/10.1088/1742-6596/2161/1/012065.

J. H. Stock and M. W. Watson. Dynamic factor models, factor-augmented vector autoregressions, and structural vector autoregressions in macroeconomics. In John B. Taylor and Harald Uhlig, editors, *Handbook of Macroeconomics*, volume 2, pages 415–525. Elsevier, 2016. DOI: 10.1016/bs.hesmac.2016.04.002.

Diego José Torres Torres. Eurozona: Evaluando la capacidad predictiva del midas. Technical Report WP15/16, BBVA Research, 2015. URL

https://www.bbvaresearch.com/publicaciones/eurozona-evaluando-la-capacidad-predictiva-del-midas/. Accedido: 2025-09-20.

Halbert White. A reality check for data snooping. *Econometrica*, 68 (5):1097–1126, 2000. DOI: https://doi.org/10.1111/1468-0262.00152. URL https://onlinelibrary.wiley.com/doi/abs/10.1111/1468-0262.00152.