
# Pronóstico Probabilístico Basado en Predicción Conformal para Series Temporales: Comparación con Bootstrapping y DeepAR

**Pedro José Leal Mesa**

Universidad Nacional de Colombia
Facultad de Ciencias, Departamento de Estadística
Bogotá D.C., Colombia
2026

# Pronóstico Probabilístico Basado en Predicción Conformal para Series Temporales: Comparación con Bootstrapping y DeepAR

**Pedro José Leal Mesa**

Tesis presentada como requisito parcial para optar al título de:

**Magíster en Ciencias - Estadística**

Director:
Ph.D. Mario E. Arrieta-Prieto

Línea de Investigación:
Análisis de Series de Tiempo y Predicción Conformal

Universidad Nacional de Colombia
Facultad de Ciencias, Departamento de Estadística
Bogotá D.C., Colombia
2026

# Dedicatoria

**A mi papá Antonio Leal, a mis compañeros y a mi director por su guía y apoyo constante en este proceso de formación.**

*"Essentially, all models are wrong, but some are useful."*

— **George E. P. Box**

# Agradecimientos

Al Ph.D. Mario E. Arrieta-Prieto, director de esta tesis, por su invaluable orientación y paciencia. Quiero agradecerle especialmente por haberme incentivado a salir de mi zona de confort y presentar los resultados preliminares de este trabajo en el escenario internacional.

Un agradecimiento profundo y especial por la oportunidad de participar en el **45th International Symposium on Forecasting** en China. Esta experiencia marcó un hito en mi carrera profesional y fue posible gracias a un esfuerzo colectivo que jamás olvidaré: a la Universidad Nacional de Colombia por brindarme los fondos institucionales, y muy especialmente a mis compañeros, amigos, conocidos y familiares, quienes con un cariño inmenso organizaron y participaron en una rifa para completar el dinero necesario para este viaje. Su solidaridad fue el motor que me llevó al otro lado del mundo.

A mis compañeros de la Maestría en Estadística, con quienes compartí este camino académico, por las discusiones técnicas y el apoyo moral en los momentos de mayor reto.

A mi padre, Antonio Leal, por ser mi apoyo incondicional y por creer en este proyecto desde el primer día. Todo este esfuerzo es también suyo.

**Con gratitud,**
**Pedro José Leal Mesa**

# Contents

# List of Figures

# List of Tables

# 1 Introducción y Objetivos

En este primer capítulo se describe el contexto general de la investigación, enfocándose en la importancia de la cuantificación de la incertidumbre en el análisis de series temporales. Se presenta el planteamiento del problema, los retos actuales de la predicción probabilística y la justificación académica de este trabajo. Finalmente, se definen los objetivos general y específicos que guiarán el desarrollo de la tesis y se presenta la estructura general del documento.

## 1.1 Introducción

La predicción es una tarea fundamental en diversas disciplinas, siendo de particular interés en el análisis de series de tiempo. Tradicionalmente, la literatura y la práctica se han centrado en estimaciones puntuales, como la media o el valor esperado de una variable futura. Sin embargo, este enfoque omite información crucial sobre la incertidumbre asociada a la predicción. Problemas clásicos, como la optimización de inventarios o la gestión de riesgos financieros, ilustran la necesidad de ir más allá de la media y considerar la distribución completa de los posibles resultados futuros para una toma de decisiones óptima.

La predicción probabilística aborda esta necesidad proporcionando no solo un valor central, sino una distribución de probabilidad o un conjunto de cuantiles sobre los valores futuros. En este contexto, surge la teoría de la *Predicción Conformal* (Conformal Prediction - CP) (Vovk, Gammerman, and Shafer 2005) como un marco robusto para la construcción de regiones de predicción con garantías de cobertura válidas. Sin embargo, es importante destacar que la CP fue desarrollada originalmente para datos idéntica e independientemente distribuidos (i.i.d.), es decir, observaciones sin ningún tipo de correlación temporal. Además, en su concepción inicial, la CP se enfoca principalmente en la construcción de intervalos de predicción en lugar de distribuciones predictivas completas. A diferencia de otros métodos, la CP posee validez exacta bajo los supuestos de inter-

cambiabilidad y es aplicable a diversos modelos predictivos, lo que la convierte en una alternativa metodológica prometedora para extender hacia el pronóstico probabilístico en series temporales.

La relevancia comparativa de los métodos que se analizarán en este trabajo radica en sus características distintivas: mientras que la CP ofrece garantías teóricas de cobertura bajo supuestos mínimos en escenarios i.i.d., los métodos de remuestreo como el *Bootstrapping* (Lahiri 2003) proporcionan flexibilidad no paramétrica para capturar la estructura de dependencia temporal, y los modelos de aprendizaje profundo como *DeepAR* (Salinas et al. 2020) destacan por su capacidad de modelar dependencias temporales complejas en múltiples series simultáneamente. Esta diversidad metodológica permite una evaluación integral de las fortalezas y limitaciones de cada enfoque.

La aplicación de CP a series de tiempo presenta desafíos críticos, ya que la dependencia temporal y los desplazamientos distribucionales suelen invalidar el supuesto de intercambiabilidad. Además, la transición de CP desde la generación de intervalos hacia distribuciones predictivas completas sigue siendo un área poco explorada. Este trabajo propone adaptar y extender estos métodos para manejar la no-intercambiabilidad y permitir la predicción distribucional, contrastando su desempeño con técnicas establecidas. Con ello, se busca cerrar una brecha relevante en la literatura: la falta de una evaluación sistemática de CP aplicada a series temporales para la obtención de distribuciones predictivas integrales.

## 1.2 Planteamiento del problema

La predicción en series de tiempo enfrenta retos fundamentales que motivan esta investigación. En primer lugar, la naturaleza secuencial de los datos introduce dependencia temporal que viola el supuesto de independencia fundamental en muchos métodos estadísticos estándar, incluyendo la predicción conformal en su formulación original. Esta dependencia temporal implica que las observaciones consecutivas están correlacionadas, lo que afecta tanto la validez de las garantías teóricas como la eficiencia de los métodos de cuantificación de incertidumbre. En segundo lugar, las características de los datos pueden cambiar con el tiempo a través de tendencias, estacionalidad o cambios estructurales, manifestando no estacionariedad que complica el modelado a largo plazo y desafía los supuestos de intercambiabilidad requeridos por métodos como la predicción conformal. Finalmente,

generar pronósticos probabilísticos que sean simultáneamente precisos (bien calibrados) e informativos (eficientes o nítidos) representa una tarea compleja, especialmente cuando se busca construir distribuciones predictivas completas en lugar de simples intervalos de predicción.

Los métodos actuales presentan limitaciones que este estudio busca contrastar. Por un lado, los modelos clásicos de tipo autorregresivo dependen fuertemente de supuestos sobre la estructura del proceso y la distribución del ruido (usualmente Gaussiano). Por otro lado, técnicas de remuestreo como el *Bootstrapping* (Lahiri 2003), aunque flexibles para capturar dependencia temporal, pueden tener dificultades cerca de límites de no estacionariedad. Finalmente, modelos modernos de aprendizaje profundo como *DeepAR* (Salinas et al. 2020) ofrecen un alto poder predictivo y la capacidad de generar distribuciones predictivas completas, pero a menudo operan como "cajas negras" y carecen de las garantías teóricas formales de cobertura que ofrece el enfoque conformal bajo condiciones de intercambiabilidad.

El problema central de este trabajo se resume en la siguiente pregunta: **¿Cómo adaptar y evaluar un modelo de predicción probabilística basado en la teoría conformal para series de tiempo que considere la dependencia temporal y permita la construcción de distribuciones predictivas completas, y cuál es su desempeño comparativo frente a métodos de referencia como Bootstrapping y DeepAR?**

## 1.3 Justificación

La cuantificación precisa de la incertidumbre es esencial para la toma de decisiones informada en áreas como la economía, meteorología y planificación de recursos. Adaptar la Predicción Conformal al dominio temporal representa un avance metodológico importante, ya que ofrece garantías de cobertura en muestra finita y una aplicabilidad general (libre de distribución) que originalmente fueron desarrolladas para el caso i.i.d.

A pesar de los desarrollos recientes en predicción conformal, existe un vacío notable en la literatura respecto a su aplicación sistemática en series temporales con características de no intercambiabilidad. La brecha metodológica que este trabajo busca cerrar es doble: por un lado, la mayoría de los trabajos en CP se han centrado en la construcción de intervalos de predicción bajo supuestos de intercambiabilidad, relegando el desarrollo de métodos para predicción distribucional completa en contextos temporales. Por otro lado, la

comparación rigurosa de CP adaptada mediante ponderación temporal frente a métodos establecidos y de vanguardia utilizando métricas probabilísticas apropiadas ha sido escasamente explorada. En particular, la evaluación mediante el promedio del *Continuous Ranked Probability Score* (ECRPS) y pruebas de significancia estadística como el test de Diebold-Mariano permitirán establecer comparaciones robustas entre las predicciones de los métodos.

La comparación sistemática con métodos estándar (Bootstrapping), estado del arte (*DeepAR*) y otros métodos proporcionará una perspectiva clara sobre las fortalezas y debilidades del enfoque conformal en escenarios reales y simulados, facilitando la selección informada de métodos de pronóstico probabilístico en aplicaciones prácticas.

## 1.4 Objetivos

### 1.4.1 Objetivo General

Formular un modelo fundamentado en la teoría conformal para realizar predicciones probabilísticas en el contexto de series de tiempo, evaluando su desempeño mediante una comparación con modelos establecidos como Bootstrapping y DeepAR.

### 1.4.2 Objetivos Específicos

- Desarrollar un modelo de predicción probabilística que integre la teoría de predicción conformal con técnicas de modelado de series temporales.
- Diseñar un entorno de simulación para evaluar el comportamiento del modelo propuesto en diversos escenarios de series temporales, incorporando estructuras temporales y condiciones de ruido variadas.
- Comparar el desempeño del modelo propuesto con modelos de referencia como DeepAR y Bootstrapping en cada escenario simulado, utilizando métricas enfocadas en predicciones probabilísticas.
- Implementar la metodología en un caso de estudio con datos reales, cuantificando su desempeño y relevancia práctica para aplicaciones de pronóstico.

## 1.5 Estructura de la tesis

El presente documento se organiza de la siguiente manera:

**Capítulo 2: Predicción Conformal.** Se presenta una introducción al pronóstico probabilístico, se explican las principales métricas para evaluar el desempeño de los modelos (incluyendo CRPS, calibración y nitidez), se desarrollan los fundamentos de los sistemas de predicción conformal, y se incluye una discusión sobre la extensión de la demostración de consistencia universal en escenarios estacionarios y ergódicos.

**Capítulo 3: Metodología de Simulación.** Se detalla el diseño del entorno de simulación, incluyendo la definición de los procesos generadores de datos, la formulación de los modelos predictivos, los cambios necesarios para su aplicación en este trabajo, y las extensiones de las simulaciones para diferentes escenarios.

**Capítulo 4: Resultados de Simulación.** Se exponen y analizan los hallazgos derivados del marco experimental descrito en el capítulo anterior. El análisis incluye la comparación del desempeño mediante la métrica ECRPS, el examen del comportamiento de modelos específicos ante diversos escenarios y la validación de los resultados a través de pruebas de significancia estadística empleando el test de Diebold-Mariano.

**Capítulo 5: Aplicaciones a Datos Reales.** Se implementan los métodos desarrollados en casos de estudio con datos reales, se presentan los resultados obtenidos y se discute su relevancia práctica.

**Capítulo 6: Conclusiones.** Se sintetizan los hallazgos principales del trabajo, se discuten las limitaciones del estudio, y se proponen direcciones para investigación futura.

# 2 Fundamentos de Pronóstico Probabilístico y Sistemas de Predicción Conformal

En este capítulo se presentan los fundamentos teóricos que sustentan el desarrollo de esta investigación. Se inicia con una introducción al pronóstico probabilístico y su importancia en el análisis de series temporales, seguido de una discusión detallada sobre las métricas utilizadas para evaluar el desempeño predictivo. Posteriormente, se desarrollan los conceptos fundamentales de la predicción conformal y sus adaptaciones para series de tiempo.

## 2.1 Pronóstico Probabilístico

El pronóstico probabilístico representa un cambio de paradigma fundamental en la predicción estadística, pasando de estimaciones puntuales a distribuciones de probabilidad completas sobre cantidades futuras de interés (Gneiting and Katzfuss 2014). A diferencia de las predicciones puntuales tradicionales, que proporcionan únicamente un valor esperado o una estimación central, el pronóstico probabilístico cuantifica la incertidumbre asociada a la predicción mediante la especificación de una distribución predictiva completa (Gneiting, Balabdaoui, and Raftery 2007).

### 2.1.1 Definición y Objetivos

Formalmente, sea $Y_{t+h}$ una variable aleatoria que representa el valor de una serie temporal en el tiempo $t + h$, donde $h > 0$ denota el horizonte de predicción. Un pronóstico probabilístico es una distribución de probabilidad $F_{t+h|t}$ que caracteriza la incertidumbre

sobre $Y_{t+h}$ dado el conjunto de información disponible hasta el tiempo $t$, denotado por $\mathcal{F}_t$ (Gneiting and Katzfuss 2014).

Gneiting y Raftery explican que el objetivo fundamental del pronóstico probabilístico es maximizar la nitidez de las distribuciones predictivas sujeto a calibración. Estos dos conceptos son fundamentales para entender la calidad de un pronóstico probabilístico (Gneiting and Raftery 2007; Gneiting, Balabdaoui, and Raftery 2007):

- **Calibración:** Se refiere a la consistencia estadística entre las distribuciones predictivas y las observaciones. Una predicción está calibrada si las realizaciones son estadísticamente indistinguibles de muestras aleatorias de las distribuciones predictivas (Thorarinsdottir and Schuhen 2017).
- **Nitidez:** Se refiere a la concentración de las distribuciones predictivas y es una propiedad exclusiva de los pronósticos. Cuanto más concentradas sean las distribuciones predictivas, mejor, siempre que se mantenga la calibración (Gneiting and Raftery 2007).

La Figura **2-1** ilustra la diferencia entre una predicción puntual, un intervalo de predicción y una distribución predictiva completa.

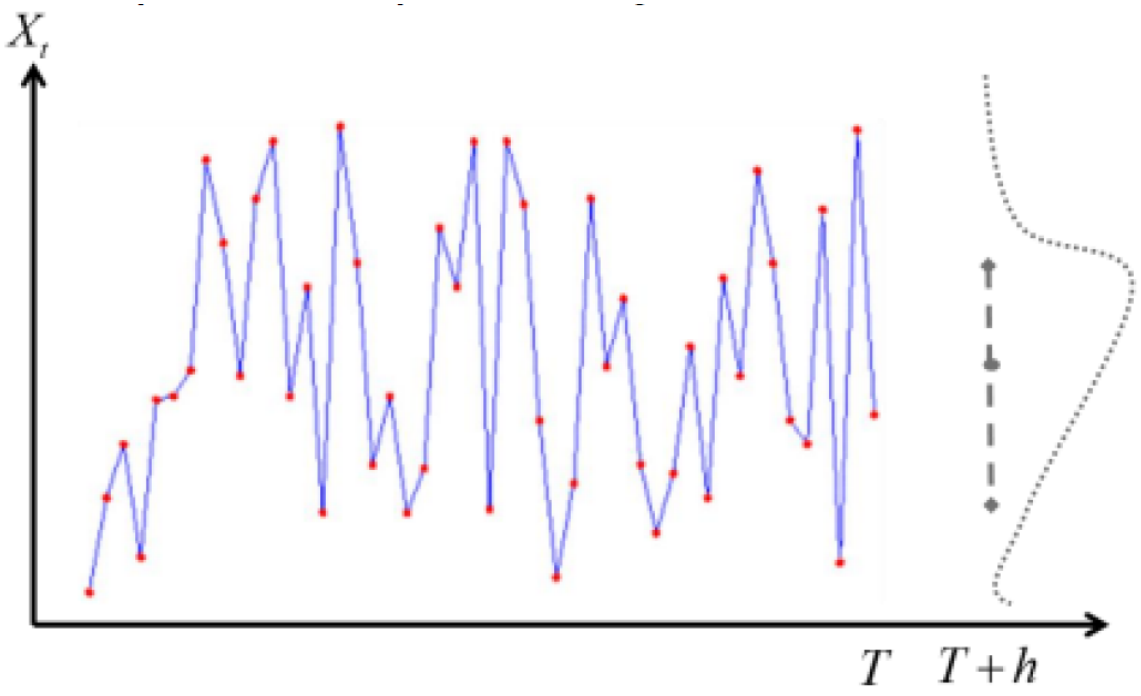


Figure **2-1**: Tipos de predicciones y su relación con la incertidumbre.

### 2.1.2 Ventajas del Pronóstico Probabilístico

El pronóstico probabilístico ofrece múltiples ventajas sobre las predicciones puntuales tradicionales que justifican su adopción creciente en diversas aplicaciones (Gneiting and Katzfuss 2014):

1. **Cuantificación Completa de Incertidumbre:** A diferencia de las predicciones puntuales, que proporcionan únicamente un valor esperado, el pronóstico probabilístico caracteriza la incertidumbre de manera exhaustiva mediante distribuciones de probabilidad. Esto permite a los tomadores de decisiones evaluar tanto la magnitud esperada de un evento como la variabilidad asociada, facilitando una comprensión más profunda de los riesgos y oportunidades (Gneiting and Raftery 2007).
2. **Soporte para Decisiones Óptimas:** En contextos donde las decisiones deben tomarse bajo incertidumbre, como la gestión de inventarios, la planificación de recursos energéticos, o la asignación de capital, las distribuciones predictivas completas son esenciales. Permiten la optimización de funciones de utilidad esperada y la implementación de estrategias que consideren explícitamente el trade-off entre riesgo y recompensa (Gneiting and Katzfuss 2014).
3. **Evaluación de Eventos Extremos:** Las predicciones puntuales son inherentemente limitadas para caracterizar eventos raros o de cola. El pronóstico probabilístico, en cambio, permite estimar probabilidades de ocurrencia de eventos extremos, información crucial para la gestión de riesgos financieros y la planificación de infraestructura (Thorarinsdottir and Schuhen 2017).
4. **Flexibilidad en la Comunicación de Incertidumbre:** Las distribuciones predictivas permiten múltiples formas de comunicación adaptadas a diferentes audiencias: intervalos de predicción, probabilidades de excedencia de umbrales críticos o visualizaciones completas mediante fan charts (Gneiting and Katzfuss 2014).

Estas ventajas han motivado la transición hacia pronósticos probabilísticos en campos tan diversos como meteorología, finanzas, energía, epidemiología y gestión de cadenas de suministro (Gneiting and Katzfuss 2014; Salinas et al. 2020).

## 2.2 Métricas para Evaluación de Pronósticos Probabilísticos

La evaluación rigurosa del desempeño predictivo es fundamental para comparar metodologías de pronóstico y guiar mejoras en los modelos. En el contexto de pronósticos probabilísticos, las métricas de evaluación deben considerar tanto la calibración como la nitidez de las distribuciones predictivas (Gneiting, Balabdaoui, and Raftery 2007; Thorarinsdottir and Schuhen 2017).

### 2.2.1 Reglas de Puntuación Propias

Una *regla de puntuación* (scoring rule) es una función $S : \mathcal{F} \times \mathbb{R} \to \mathbb{R} \cup \{\infty\}$ que asigna una penalización numérica $S(F, y)$ a cada par formado por una distribución predictiva $F$ y una observación realizada $y$ (Gneiting and Raftery 2007). En nuestra notación, valores más bajos de la puntuación indican mejor desempeño predictivo.

#### Propriety y Strict Propriety

La *propriety* es una característica fundamental que debe satisfacer toda métrica de evaluación de pronósticos probabilísticos para garantizar que incentive predicciones honestas y bien calibradas (Gneiting and Raftery 2007).

**Definición (Regla de Puntuación Proper):** Una regla de puntuación $S$ es *proper* relativa a una clase $\mathcal{F}$ de distribuciones de probabilidad si

$$\mathbb{E}_G[S(G, Y)] \leq \mathbb{E}_G[S(F, Y)] \tag{2-1}$$

para todas las distribuciones $F, G \in \mathcal{F}$, donde $Y \sim G$ (Gneiting and Raftery 2007; Thorarinsdottir and Schuhen 2017).

**Definición (Regla de Puntuación Strictly Proper):** La regla de puntuación $S$ es *strictly proper* si la desigualdad en (2-1) se cumple con igualdad únicamente cuando $F = G$ (Gneiting and Raftery 2007).

La importancia de la *propriety* radica en que establece un principio de alineación de incentivos: si un pronosticador desea minimizar su puntuación esperada, su mejor estrategia es

reportar sinceramente su verdadera distribución predictiva (Gneiting and Raftery 2007).

### 2.2.2 Continuous Ranked Probability Score (CRPS)

El *Continuous Ranked Probability Score* (CRPS) es una de las reglas de puntuación estrictamente propias más utilizadas para evaluar pronósticos probabilísticos de variables continuas (Gneiting and Katzfuss 2014). Su popularidad se debe a su sólida fundamentación teórica y su capacidad para evaluar simultáneamente calibración y nitidez (Gneiting and Raftery 2007; Thorarinsdottir and Schuhen 2017).

#### Definiciones y Representaciones

El CRPS admite varias representaciones matemáticas equivalentes, cada una con sus propias ventajas conceptuales y computacionales.

**Representación integral:** La definición original del CRPS está dada por (Gneiting and Raftery 2007):

$$\text{CRPS}(F, y) = \int_{-\infty}^{\infty} \left(F(x) - \mathbb{1}\{y \leq x\}\right)^2 dx \tag{2-2}$$

donde $F$ es la función de distribución acumulada (FDA) de la distribución predictiva y $y$ es la observación realizada. Esta representación muestra que el CRPS mide el área entre la FDA predictiva y la FDA de la observación (Thorarinsdottir and Schuhen 2017).

**Representación basada en esperanzas:** Una forma alternativa, más conveniente para cálculos, está dada por (Gneiting and Raftery 2007):

$$\text{CRPS}(F, y) = \mathbb{E}_F|X - y| - \frac{1}{2}\mathbb{E}_F|X - X'| \tag{2-3}$$

donde $X$ y $X'$ son variables aleatorias independientes con distribución $F$. Esta representación revela una interpretación intuitiva del CRPS: el primer término mide la distancia esperada entre la predicción y la observación, mientras que el segundo término penaliza la dispersión de la distribución predictiva.

#### Propiedades del CRPS

El CRPS posee varias propiedades deseables que explican su amplia adopción en la literatura (Gneiting and Raftery 2007; Thorarinsdottir and Schuhen 2017):

1. **Strictly proper:** El CRPS es *strictly proper* relativo a la clase de todas las distribuciones de probabilidad en $\mathbb{R}$ con primer momento finito (Gneiting and Raftery 2007).
2. **Unidades consistentes:** El CRPS se expresa en las mismas unidades que la variable pronosticada (Gneiting and Katzfuss 2014).
3. **Reducción al error absoluto:** Cuando $F$ es una distribución degenerada (predicción puntual), el CRPS se reduce al error absoluto $|x - y|$, permitiendo un marco de evaluación unificado (Gneiting and Raftery 2007).
4. **Sensibilidad dual:** El CRPS evalúa simultáneamente la calibración y la nitidez (Thorarinsdottir and Schuhen 2017).

### 2.2.3 Expected Continuous Ranked Probability Score (ECRPS)

El desempeño predictivo global de una secuencia de $n$ pares pronóstico-observación se cuantifica mediante el *Expected Continuous Ranked Probability Score* (ECRPS), definido como la media aritmética de los CRPS individuales:

$$\text{ECRPS} = \frac{1}{n} \sum_{i=1}^{n} \text{CRPS}(F_i, y_i) \tag{2-4}$$

El ECRPS hereda todas las propiedades deseables del CRPS y proporciona un resumen numérico único del desempeño predictivo sobre todo el conjunto de evaluación. En particular, la comparación mediante ECRPS permite establecer comparaciones robustas entre diferentes métodos de pronóstico probabilístico.

## 2.3 Test de Diebold-Mariano para Comparación de Precisión Predictiva

La evaluación comparativa de distintas metodologías de pronóstico requiere herramientas estadísticas rigurosas que permitan determinar si las diferencias observadas en el desempeño predictivo son estadísticamente significativas o simplemente producto del azar. El test de Diebold-Mariano (Diebold and Mariano 1995) constituye uno de los procedimientos más ampliamente utilizados para este propósito, ofreciendo un marco general y flexible para contrastar la hipótesis nula de igual precisión predictiva entre dos métodos de pronóstico competidores.

### 2.3.1 Formulación del Test

Sean $\hat{y}_{t+h}^{(1)}$ y $\hat{y}_{t+h}^{(2)}$ dos pronósticos $h$ pasos adelante para una variable $y_{t+h}$, producidos por dos metodologías diferentes. Los errores de pronóstico correspondientes son:

$$e_{t+h}^{(i)} = y_{t+h} - \hat{y}_{t+h}^{(i)}, \quad i = 1, 2 \tag{2-5}$$

El test de Diebold-Mariano se basa en una función de pérdida $L(\cdot)$ que cuantifica el costo asociado con cada error de pronóstico. Para un horizonte temporal de evaluación que abarca $n$ observaciones, se define el diferencial de pérdida en el tiempo $t$ como:

$$d_t = L(e_t^{(1)}) - L(e_t^{(2)}), \quad t = 1, \ldots, n \tag{2-6}$$

Tradicionalmente, el test de Diebold-Mariano se ha aplicado utilizando la pérdida cuadrática para evaluar estimaciones puntuales. Sin embargo, este marco es suficientemente general para acomodar cualquier función de pérdida (Diebold and Mariano 1995). En el presente trabajo, se utilizará principalmente el CRPS o el ECRPS, según corresponda como métrica de pérdida fundamental. Esto permite extender la comparación de Diebold-Mariano.

La hipótesis nula de igual precisión predictiva se formula como:

$$H_0 : \mathbb{E}[d_t] = 0 \tag{2-7}$$

Esta hipótesis establece que la pérdida esperada es idéntica para ambos métodos de pronóstico. El estadístico de prueba se construye a partir de la media muestral del diferencial de pérdida:

$$\bar{d} = \frac{1}{n} \sum_{t=1}^{n} d_t \tag{2-8}$$

### 2.3.2 Distribución Asintótica y Estimación de la Varianza

Bajo condiciones de regularidad que incluyen la estacionariedad débil y la existencia de momentos de orden finito, Diebold y Mariano demuestran que:

$$\sqrt{n}\,\bar{d} \xrightarrow{d} N(0, 2\pi f_d(0)) \tag{2-9}$$

donde $f_d(0)$ denota la densidad espectral de la serie $d_t$ evaluada en frecuencia cero, la cual equivale a la varianza de largo plazo:

$$\sigma^2 = \mathrm{Var}(\sqrt{n}\,\bar{d}) = \gamma_0 + 2\sum_{k=1}^{\infty} \gamma_k \tag{2-10}$$

siendo $\gamma_k = \mathrm{Cov}(d_t, d_{t-k})$ la autocovarianza de orden $k$.

Un aspecto fundamental del test de Diebold-Mariano es que permite explícitamente la presencia de autocorrelación en el diferencial de pérdida $d_t$. Esta característica es especialmente relevante en el contexto de pronósticos a múltiples pasos adelante ($h > 1$), donde los errores de pronóstico exhiben típicamente estructura de autocorrelación hasta el orden $(h-1)$ (Diebold and Mariano 1995). Esta estructura surge porque pronósticos óptimos $h$ pasos adelante generan errores que siguen un proceso de media móvil MA$(h-1)$.

En la práctica, la varianza de largo plazo $\sigma^2$ debe ser estimada. Diebold y Mariano proponen utilizar un estimador basado en autocovarianzas ponderadas por kernel:

$$\hat{\sigma}^2 = \hat{\gamma}_0 + 2\sum_{k=1}^{M} k\left(\frac{k}{M}\right)\hat{\gamma}_k \tag{2-11}$$

donde $\hat{\gamma}_k = n^{-1}\sum_{t=k+1}^{n}(d_t - \bar{d})(d_{t-k} - \bar{d})$ son las autocovarianzas muestrales, $k(\cdot)$ es una función kernel (por ejemplo, kernel de Bartlett o Parzen), y $M$ es el parámetro de ancho de banda o truncamiento que controla el número de autocovarianzas incluidas en la

estimación.

Para el caso específico donde se conoce que el diferencial de pérdida sigue un proceso MA($h-1$), Diebold y Mariano sugieren simplificar el estimador utilizando $M = h-1$ con kernel rectangular:

$$\hat{\sigma}^2_{DM} = \hat{\gamma}_0 + 2\sum_{k=1}^{h-1} \hat{\gamma}_k \tag{2-12}$$

El estadístico de prueba resultante es:

$$DM = \frac{\sqrt{n}\,\bar{d}}{\hat{\sigma}} \tag{2-13}$$

Bajo la hipótesis nula, este estadístico converge en distribución a una normal estándar: $DM \xrightarrow{d} N(0,1)$. Para un test bilateral al nivel de significancia $\alpha$, se rechaza $H_0$ si $|DM| > z_{\alpha/2}$, donde $z_{\alpha/2}$ denota el cuantil $(1-\alpha/2)$ de la distribución normal estándar.

### 2.3.3 Modificaciones para Muestras Pequeñas

A pesar de la solidez teórica del test de Diebold-Mariano bajo asintótica estándar, diversos estudios han documentado distorsiones de tamaño en muestras finitas, particularmente cuando el número de observaciones de pronóstico es limitado. Harvey et al. (Harvey, Leybourne, and Newbold 1997) demostraron mediante simulaciones Monte Carlo que el test original tiende a sobrerrechazar la hipótesis nula (es decir, presenta un tamaño empírico superior al nominal), especialmente para horizontes de pronóstico largos y muestras pequeñas.

Para abordar estas limitaciones, Harvey et al. proponen una corrección del estadístico que mejora sustancialmente el desempeño en muestras finitas. La modificación se fundamenta en el uso de un estimador aproximadamente insesgado de la varianza de $\bar{d}$. Partiendo de la expresión exacta:

$$\mathrm{Var}(\bar{d}) = n^{-1}\left[\gamma_0 + 2n^{-1}\sum_{k=1}^{h-1}(n-k)\gamma_k\right] \tag{2-14}$$

y calculando el valor esperado del estimador empleado en (2-12), se obtiene que:

$$\mathbb{E}[\hat{\sigma}^2_{DM}] \approx \left[\frac{n+1-2h+n^{-1}h(h-1)}{n}\right]\mathrm{Var}(\bar{d}) \tag{2-15}$$

Esta relación sugiere el estadístico modificado:

$$DM^* = \left[\frac{n + 1 - 2h + n^{-1}h(h-1)}{n}\right]^{1/2} DM \tag{2-16}$$

Adicionalmente, Harvey et al. recomiendan comparar $DM^*$ con valores críticos de la distribución $t$ de Student con $(n-1)$ grados de libertad, en lugar de la distribución normal estándar. Esta segunda modificación reconoce implícitamente la incertidumbre adicional asociada con la estimación de la varianza en muestras finitas.

Los resultados de simulación reportados por Harvey et al. (Harvey, Leybourne, and Newbold 1997) indican que el test modificado presenta un tamaño empírico considerablemente más cercano al nominal, especialmente para $n \leq 50$ y horizontes de pronóstico $h \geq 2$. Aunque el test modificado exhibe una ligera pérdida de potencia en comparación con el test original cuando ambos están correctamente calibrados, esta reducción es marginal y ampliamente compensada por la ganancia en confiabilidad inferencial.

### 2.3.4 Enfoque de Asintótica de Suavizado Fijo

Una alternativa más reciente para abordar las distorsiones de tamaño del test de Diebold-Mariano en muestras pequeñas es el enfoque de *asintótica de suavizado fijo* (fixed-smoothing asymptotics), desarrollado por Coroneo e Iacone (Coroneo and Iacone 2020). Este marco teórico reconoce que en aplicaciones prácticas de evaluación de pronósticos, el tamaño muestral $n$ es frecuentemente limitado, haciendo que la aproximación asintótica estándar (que requiere $M/n \to 0$) sea inadecuada.

La idea fundamental es mantener constante la razón entre el parámetro de ancho de banda y el tamaño muestral conforme $n$ aumenta. Formalmente, bajo la *asintótica fixed-b*, se asume que $M/n \to b$ para algún $b \in (0, 1]$ fijo. Bajo este régimen asintótico alternativo, el estimador de varianza (2-11) ya no es consistente para $\sigma^2$. Sin embargo, Kiefer y Vogelsang (2005) demostraron que el estadístico resultante converge a una distribución no estándar que depende de $b$ y del kernel empleado.

Para el kernel de Bartlett, la distribución límite puede caracterizarse explícitamente, y sus cuantiles pueden aproximarse mediante fórmulas polinomiales. Específicamente, para un

test bilateral al 5% de significancia, el valor crítico $c_\alpha(b)$ satisface:

$$c_\alpha(b) \approx \alpha_0 + \alpha_1 b + \alpha_2 b^2 + \alpha_3 b^3 \tag{2-17}$$

donde los coeficientes $\{\alpha_i\}$ han sido tabulados por Kiefer y Vogelsang.

Coroneo e Iacone (Coroneo and Iacone 2020) extienden este marco al contexto específico de evaluación de pronósticos, demostrando mediante simulaciones Monte Carlo que los tests basados en asintótica de suavizado fijo exhiben un tamaño empírico notablemente más preciso que el test de Diebold-Mariano estándar, incluso para muestras tan pequeñas como $n = 40$. Los autores proponen utilizar anchos de banda $M = \lfloor n^{1/2} \rfloor$ para el estimador con kernel de Bartlett, encontrando que esta elección ofrece un equilibrio favorable entre tamaño y potencia del test.

Una segunda variante dentro del paradigma de suavizado fijo es la *asintótica fixed-m*, que emplea un estimador de varianza basado en el periodograma suavizado con kernel de Daniell:

$$\hat{\sigma}^2_{DAN} = \frac{2\pi}{m} \sum_{j=1}^{m} I(\lambda_j) \tag{2-18}$$

donde $I(\lambda_j)$ denota el periodograma de $d_t$ evaluado en la frecuencia de Fourier $\lambda_j = 2\pi j/n$, y $m$ es un parámetro de truncamiento mantenido fijo conforme $n \to \infty$. Bajo condiciones de regularidad, el estadístico resultante converge a una distribución $t$ con $2m$ grados de libertad (Coroneo and Iacone 2020).

### 2.3.5 Consideraciones sobre la Generación de Pronósticos

Un aspecto metodológico crucial del test de Diebold-Mariano es su tratamiento de los pronósticos como objetos dados o primitivos, sin considerar explícitamente el proceso de estimación de los modelos subyacentes. Esta perspectiva contrasta con marcos alternativos, como el de West (1996) y Clark y McCracken (2001), que desarrollan teoría asintótica específica para pronósticos derivados de modelos paramétricos estimados.

Cuando los pronósticos provienen de modelos con parámetros estimados, la incertidumbre asociada con la estimación puede afectar la distribución del estadístico de prueba. West (West 1996) demostró que, bajo ciertas condiciones, esta incertidumbre de estimación es asintóticamente despreciable si el tamaño de la muestra de entrenamiento $R$ crece mucho

más rápido que el tamaño de la muestra de evaluación $P$ (específicamente, si $P/R \to 0$).

Alternativamente, Giacomini y White (Giacomini and White 2006) proponen un marco donde la incertidumbre de estimación no desaparece asintóticamente (fijando $R < \infty$). Bajo este régimen, el test de Diebold-Mariano permanece válido, pero ahora evalúa el desempeño relativo de *métodos de pronóstico* (incluyendo el procedimiento de estimación) en lugar de *modelos de pronóstico* poblacionales. Este marco es particularmente apropiado cuando el objetivo es comparar estrategias de pronóstico que podrían implementarse en práctica, reconociendo que los modelos deben ser reestimados periódicamente con ventanas de datos finitas.

Para los propósitos de esta investigación, adoptamos la perspectiva de Giacomini y White, interpretando el test de Diebold-Mariano como una herramienta para evaluar el desempeño predictivo de métodos completos de pronóstico, incluyendo tanto la especificación del modelo como el procedimiento de estimación y actualización de parámetros.

## 2.4 Predicción Conformal por Intervalos: El Enfoque IIE

La predicción conformal clásica, introducida por Vovk et al. Vovk, Gammerman, and Shafer 2005, se fundamenta en la capacidad de generar conjuntos de predicción $\Gamma^\epsilon$ que garantizan una cobertura de confianza exacta para cualquier nivel de significancia $\epsilon \in (0, 1)$. A diferencia de los métodos estadísticos tradicionales que dependen de la asintótica (grandes muestras), la predicción conformal es válida para muestras finitas, siempre que se cumpla el supuesto de intercambiabilidad de los datos.

### 2.4.1 El Concepto de No-Conformidad

El núcleo de esta metodología es la *medida de no-conformidad* (NCM, por sus siglas en inglés). Una NCM es una función $A(B, z)$ que cuantifica el grado de "extrañeza" de un ejemplo $z$ en relación con un multiconjunto (o *bag*) de ejemplos $B$. En el contexto de regresión, donde $z = (x, y)$, la medida de no-conformidad más común es el error absoluto de predicción, definido como:

$$\alpha_i = |y_i - \hat{y}_i| \tag{2-19}$$

donde $\hat{y}_i$ es la estimación producida por un algoritmo de aprendizaje subyacente (denominado *underlying algorithm*). Es importante subrayar que la predicción conformal es agnóstica al modelo: puede envolver desde una regresión lineal simple hasta redes neuronales profundas, transformando sus predicciones puntuales en intervalos con validez estadística.

### 2.4.2 Protocolo de Construcción de Intervalos

Para construir un intervalo de predicción para un nuevo objeto $x_n$ basado en un conjunto de entrenamiento $z_1, \ldots, z_{n-1}$, el método IIE (Inducida por Errores) sigue un proceso de prueba de hipótesis inversa. Para cada valor potencial $y \in \mathbb{R}$:

1. **Aumentación del Conjunto:** Se asume hipotéticamente que la verdadera etiqueta de $x_n$ es $y$, formando el conjunto aumentado $z_1, z_2, \ldots, z_{n-1}, z_n$, donde $z_n = (x_n, y)$.
2. **Cálculo de Puntajes:** Se calculan los puntajes de no-conformidad $\alpha_1, \ldots, \alpha_n$ para todos los elementos, incluyendo el ejemplo hipotético.
3. **Derivación del p-valor:** Se calcula la proporción de ejemplos que son "al menos tan extraños" como el nuevo ejemplo $z_n$:

$$p(y) = \frac{|\{i = 1, \ldots, n : \alpha_i \geq \alpha_n\}|}{n} \tag{2-20}$$

4. **Inversión de la Región de Aceptación:** El intervalo de predicción $\Gamma^{1-\epsilon}$ se define como el conjunto de todos los valores $y$ que no pueden ser rechazados al nivel de significancia $\epsilon$:

$$\Gamma^{\epsilon}(x_1, y_1, \ldots, x_{n-1}, y_{n-1}, x_n) = \{y \in \mathbb{R} : p(y) > \epsilon\} \tag{2-21}$$

Este procedimiento garantiza que $P(y_n \notin \Gamma^{\epsilon}) \leq \epsilon$. Si los puntajes $\alpha_i$ tienen una distribución continua (sin empates), la probabilidad de error es exactamente $\epsilon$ Vovk, Gammerman, and Shafer 2005.

## 2.5 Robustez ante la No-Intercambiabilidad: Aproximación de Barber

Uno de los desafíos críticos en el análisis de series temporales es que el supuesto de intercambiabilidad rara vez se sostiene. Fenómenos como la autocorrelación, la heterocedasticidad y la deriva de parámetros (drift) invalidan la asunción de que el pasado y el futuro son estadísticamente idénticos. Barber et al. Barber et al. 2023 proponen una extensión fundamental para estos escenarios.

### 2.5.1 El Gap de Cobertura y Variación Total

Barber et al. formalizan la degradación de la validez conformal mediante el uso de la *Distancia de Variación Total* ($d_{TV}$). Si la distribución de los datos cambia en el tiempo, existe una brecha de cobertura (*coverage gap*). El teorema principal de Barber establece que la pérdida de cobertura está acotada por la suma de las distancias entre la distribución de los datos de entrenamiento y la distribución del dato de prueba:

$$\text{Error de Cobertura} \leq \epsilon + \sum_{i=1}^{n} w_i d_{TV}(Z_i, Z_{n+1}) \tag{2-22}$$

### 2.5.2 Cuantiles Pesados y Decaimiento Temporal

Para contrarrestar este efecto en series de tiempo, Barber et al. introducen los *Weighted Conformal Predictors*. En lugar de asignar un peso uniforme de $1/n$ a cada residuo histórico, se asignan pesos $w_i$ que reflejan la relevancia del dato. En series no estacionarias, los datos más recientes son mejores predictores del futuro.

Se define comúnmente un decaimiento geométrico para los pesos:

$$w_i = \rho^{n-i}, \quad \rho \in (0, 1) \tag{2-23}$$

donde un $\rho$ cercano a 1 asume una estabilidad lenta, mientras que un $\rho$ menor reacciona rápidamente a cambios estructurales. El p-valor pesado se calcula como una suma pon-

derada de funciones indicadoras:

$$p^y = \frac{\sum_{i=1}^{n-1} w_i \mathbb{1}_{\alpha_i \geq \alpha_n} + w_n}{\sum_{j=1}^{n} w_j} \tag{2-24}$$

Este enfoque permite que la predicción conformal sea "adaptativa", manteniendo la cobertura cercana al nivel nominal incluso cuando la serie temporal experimenta cambios súbitos en su media o varianza Barber et al. 2023.

## 2.6 Sistemas de Predicción Conformal (CPS): De Intervalos a Densidades

El Capítulo 7 de la obra de Vovk Vovk, Gammerman, and Shafer 2005 marca la transición de la predicción de conjuntos a la predicción de distribuciones completas. Un *Sistema de Predicción Conformal* (CPS) no entrega un rango, sino una *Distribución Predictiva Aleatorizada* (RPD), denotada como $\Pi_n(y, \tau)$, que representa la probabilidad de que la verdadera etiqueta sea menor o igual a $y$.

### 2.6.1 Formalización de la RPD y el Suavizado ($\tau$)

Para asegurar que la distribución resultante sea continua y cumpla con las propiedades de una FDA (Función de Distribución Acumulada), se introduce una variable de suavizado $\tau \sim U(0, 1)$. La función $\Pi$ se define como:

$$\Pi_n(y, \tau) := \frac{|\{i : \alpha_i < \alpha_n^y\}| + \tau |\{i : \alpha_i = \alpha_n^y\}|}{n} \tag{2-25}$$

Es vital notar que aquí $\alpha_i$ son puntajes de *conformidad* (no de no-conformidad). Un ejemplo común en regresión es $\alpha_i = y_i - \hat{y}_i$. El uso de la variable $\tau$ garantiza la *calibración fuerte en probabilidad*: los valores de la RPD evaluados en la verdadera etiqueta son independientes y uniformes en $[0, 1]$, permitiendo una cuantificación exacta de la incertidumbre en cualquier punto de la distribución Vovk, Nouretdinov, et al. 2017.

### 2.6.2 La Máquina de Predicción de Mínimos Cuadrados (LSPM)

La *Least Squares Prediction Machine* (LSPM) es la aplicación primordial de los CPS al ámbito de la regresión. La LSPM utiliza la estructura de la regresión lineal para optimizar la eficiencia de la distribución predictiva.

#### Variantes de la LSPM

Vovk distingue tres formas de calcular los residuos dentro de una LSPM:

1. **LSPM Ordinaria:** Los puntajes son simplemente los residuos de entrenamiento. Sin embargo, este enfoque tiende a ser demasiado optimista (sobreajuste), ya que el modelo ya ha "visto" los datos de entrenamiento.
2. **LSPM Eliminada (Deleted):** Utiliza un esquema de validación cruzada interna (*leave-one-out*). Para cada dato $i$, se entrena un modelo omitiendo ese dato específico, asegurando que el residuo sea una medida honesta de la capacidad de generalización.
3. **LSPM Estudiantizada:** Es la variante más robusta y matemáticamente rigurosa. Ajusta cada residuo por su apalancamiento (*leverage*), $h_i$, proveniente de la diagonal de la matriz *hat*:

$$\alpha_i := \frac{y_i - \hat{y}_i}{\sigma\sqrt{1 - h_i}} \tag{2-26}$$

## 2.7 Sistemas de Predicción Conformal de Mondrian (MCPS)

A pesar de las sólidas garantías de validez marginal que ofrecen los Sistemas de Predicción Conformal (CPS) descritos en la sección 2.6, estos presentan una limitación teórica y práctica fundamental: la garantía de error es un promedio sobre todo el espacio de datos. Esto implica que el sistema puede ser extremadamente preciso en ciertas regiones del espacio de características y, simultáneamente, cometer errores sistemáticos en otras, siempre que el error global no supere el nivel $\epsilon$. Los *Sistemas de Predicción Conformal de Mondrian* (MCPS, por sus siglas en inglés) introducen el concepto de *validez condicional por categorías*, permitiendo que la calibración se mantenga exacta dentro de subconjuntos

específicos de los datos Vovk, Gammerman, and Shafer 2022.

### 2.7.1 Origen y Motivación: Validez Marginal vs. Condicional

El apelativo "Mondrian" deriva del estilo geométrico del pintor neerlandés Piet Mondrian, cuya estética se fundamenta en la compartimentación del lienzo en rectángulos de colores puros delimitados por una cuadrícula, tal como se ilustra en la Figura **2-2**. Bajo esta analogía, un sistema de predicción conformal Mondriano particiona el espacio de ejemplos $\mathcal{Z}$ en categorías mutuamente excluyentes o taxonomías. Este enfoque permite que las garantías de cobertura sean válidas no solo de forma agregada, sino específicamente dentro de cada subgrupo definido, abordando así el problema de la validez condicional.

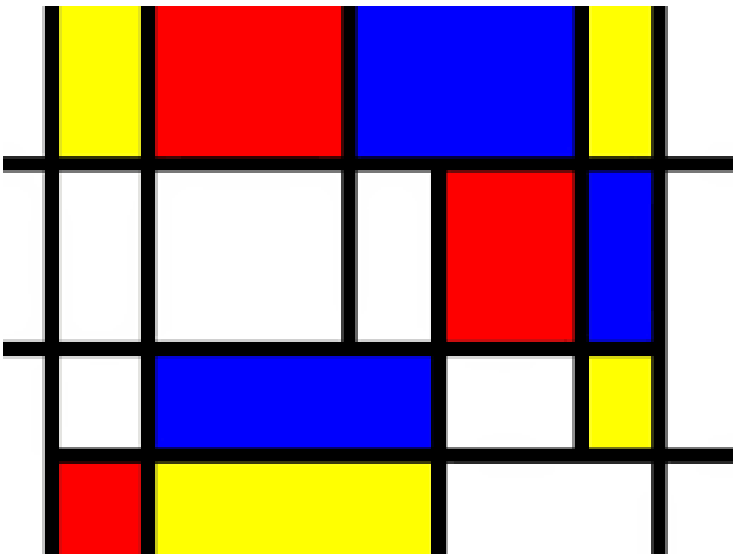

Figure **2-2**: Estética de Mondrian como analogía de la partición del espacio $\mathcal{Z}$.

La necesidad de este enfoque surge cuando existen grupos de datos con dificultades predictivas heterogéneas. Por ejemplo, en una serie temporal de demanda eléctrica, predecir el consumo en un día festivo es intrínsecamente más difícil que en un día laboral. Un CPS global podría subestimar masivamente la incertidumbre en los días festivos, compensándola con una sobreestimación en los días laborales. El enfoque de Mondrian garantiza que la probabilidad de error sea exactamente $\epsilon$ tanto para los días laborales como para los festivos, de forma independiente Vovk, Nouretdinov, et al. 2017.

### 2.7.2 La Taxonomía de Mondrian ($\kappa$)

La base matemática de un MCPS es la *taxonomía*. Una taxonomía es una función medible $\kappa : \mathbb{N} \times (\mathbf{X} \times \mathbf{Y}) \to K$, donde $K$ es un conjunto numerable de categorías. Para cada par de ejemplo $(x_i, y_i)$ y su posición en la secuencia $i$, la taxonomía asigna una categoría $\kappa_i$.

Existen tres tipos principales de taxonomías aplicables a series temporales:

1. **Taxonomías de Objetos:** Dependen solo de las características $x_i$ (ej. agrupar por niveles de volatilidad observada).
2. **Taxonomías de Etiquetas:** Dependen de la respuesta $y_i$. Esto da lugar a los *Label-Conditional Conformal Predictors*, vitales cuando el impacto de un error depende de la magnitud del valor (ej. errores en valores extremos son más costosos).
3. **Taxonomías Temporales:** Dependen del índice $i$. Este es el puente con el trabajo de Barber et al. Barber et al. 2023, donde la categoría de Mondrian puede ser una "ventana deslizante" de los datos más recientes para adaptarse a la no-intercambiabilidad.

### 2.7.3 Integración del Algoritmo MCPS

La integración de la lógica de Mondrian en un Sistema de Predicción Conformal se realiza modificando el cálculo del p-valor o de la RPD (Distribución Predictiva Aleatorizada). En lugar de comparar el puntaje del nuevo ejemplo $\alpha_n$ con todos los puntajes históricos, solo se compara con aquellos que pertenecen a su misma categoría.

Sea $\sigma = \{z_1, \ldots, z_{n-1}\}$ el conjunto de entrenamiento y $z_n = (x_n, y)$ el ejemplo de prueba con etiqueta hipotética $y$. El proceso para generar la RPD de Mondrian $\Pi_M$ es el siguiente:

1. Se identifica la categoría del nuevo ejemplo: $k = \kappa(n, (x_n, y))$.
2. Se filtran los índices de los ejemplos de entrenamiento que pertenecen a dicha categoría:
$$S_k = \{i \in \{1, \ldots, n-1\} : \kappa(i, z_i) = k\} \tag{2-27}$$
3. Se calculan los puntajes de conformidad $\alpha_i$ solo para $i \in S_k \cup \{n\}$.

4. La RPD de Mondrian se define como:

$$\Pi_M(y,\tau) := \frac{|\{i \in S_k : \alpha_i < \alpha_n^y\}| + \tau|\{i \in S_k \cup \{n\} : \alpha_i = \alpha_n^y\}|}{|S_k| + 1} \tag{2-28}$$

El denominador $|S_k| + 1$ es clave: representa el tamaño de la "muestra local". Si una categoría tiene pocos ejemplos, la distribución predictiva será naturalmente más dispersa (reflejando mayor incertidumbre), mientras que categorías ricas en datos producirán densidades más nítidas Vovk, Gammerman, and Shafer 2022.

## 2.8 Análisis de la Consistencia Universal de Vovk

Para consolidar el marco teórico de esta investigación, es imperativo discutir el sustento matemático que garantiza que los Sistemas de Predicción Conformal (CPS) no solo son válidos en muestras finitas, sino también óptimos a medida que el volumen de datos aumenta. Este respaldo proviene de la demostración de la *consistencia universal* de Vovk (Vovk 2019), formalizada en el Teorema 31 de su obra reciente.

### 2.8.1 Definición de Consistencia Universal

En el contexto de los CPS, la validez (propiedad R2) asegura que el sistema está calibrado independientemente de la distribución de los datos. Sin embargo, la validez por sí sola no garantiza que la distribución predictiva $\Pi_n$ sea una buena aproximación a la verdadera distribución condicional de las etiquetas $P(y|x)$.

Vovk define un sistema predictivo como *universalmente consistente* si, para cualquier medida de probabilidad $P$ (bajo el modelo IID) y para cualquier función continua acotada $f$, se cumple que:

$$\int f d\Pi_n - \mathbb{E}_P(f|x_{n+1}) \to 0 \quad \text{en probabilidad cuando } n \to \infty \tag{2-29}$$

Esta propiedad implica que, asintóticamente, el CPS "encuentra" la verdadera distribución de probabilidad generadora de los datos, eliminando la incertidumbre epistémica conforme el tamaño de la muestra $n$ tiende al infinito Vovk 2019.

### 2.8.2 Mecanismo de la Demostración: El Enfoque de Histograma

La prueba de Vovk sobre la existencia de un CPS universal se apoya en la construcción de un *Histogram Conformal Predictive System*. El argumento se divide en dos pilares fundamentales que vinculan la teoría de martingalas con la ley de los grandes números:

1. **Teorema de Convergencia de Martingalas de Lévy:** Vovk utiliza particiones anidadas del espacio de objetos $X$ (celdas de histograma que se encogen conforme $n$ crece). Según el teorema de Lévy, la esperanza condicional de la función sobre una celda que se reduce tiende al valor puntual de la esperanza condicional en el objeto de prueba $x_{n+1}$ Vovk 2019.
2. **Ley de los Grandes Números (LGN):** Mientras las celdas se encogen para ganar resolución, el número de ejemplos dentro de cada celda debe tender a infinito ($nh_n \to \infty$, donde $h_n$ es el ancho de la celda). Esto permite que la frecuencia empírica de las etiquetas dentro de la categoría de Mondrian converja a la esperanza real en esa región del espacio Vovk 2019.

### 2.8.3 La Distancia de Lévy y la Convergencia Débil

Un punto crítico de la demostración es el uso de la noción de Belyaev sobre secuencias de distribuciones que se aproximan débilmente. Vovk demuestra que bajo un CPS universal, la *Distancia de Lévy* entre la distribución predictiva conformal y la verdadera distribución condicional converge a cero en probabilidad Vovk 2019.

Este resultado es el que otorga rigor a la aplicación de CPS en problemas de alta criticidad, como el pronóstico de carga eléctrica o la gestión de riesgos financieros. Indica que el analista no tiene que elegir entre un modelo "seguro" (conformal) y un modelo "preciso" (bayesiando/paramétrico); el CPS universal ofrece ambas ventajas simultáneamente:

- **A corto plazo:** Garantiza cobertura exacta mediante calibración fuerte.
- **A largo plazo:** Garantiza convergencia a la distribución real de los datos sin requerir asunciones paramétricas.

### 2.8.4 Implicaciones para el LSPM y la Eficiencia

Aunque el modelo de mínimos cuadrados (LSPM) estudiado en la sección 2.6 es eficiente bajo ruido gaussiano, Vovk advierte que no es universalmente consistente si la relación real entre $X$ y $Y$ no es lineal Vovk, Gammerman, and Shafer 2005. Por ello, el desarrollo de CPS basados en kernels o en métodos de vecinos cercanos (como se discute en el capítulo 4 de su obra) es lo que permite alcanzar la consistencia universal en espacios de características complejos. Esta conclusión justifica el uso de arquitecturas no lineales conformizadas en la presente tesis, ya que heredan la solidez de la prueba de consistencia de Vovk.

## 2.9 Hacia la Consistencia Universal en Series de Tiempo Ergódicas

Si bien el trabajo de Vovk (Vovk 2019) establece una base sólida para la consistencia universal bajo el modelo IID, las aplicaciones en entornos reales, como las series de tiempo, exigen una transición hacia modelos que capturen la dependencia temporal. Inspirado en el formalismo de Vovk, el presente marco teórico propone las bases para un Sistema de Predicción Conformal (CPS) adaptado a procesos estocásticos donde la suposición de intercambiabilidad no se cumple.

### 2.9.1 Redefinición del Objetivo de Consistencia

En el contexto de series de tiempo, el objetivo de un CPS universalmente consistente es que la distribución predictiva generada, $Q_n(y)$, converja débilmente en probabilidad a la verdadera distribución condicional $F_{Y|X}(\cdot|X_{n+1})$. A diferencia del caso IID, aquí la "consistencia" implica que el sistema debe ser capaz de aprender la dinámica local y la estructura de dependencia del proceso a medida que la serie evoluciona.

Se plantea que, bajo este régimen, el sistema no solo debe ser asintóticamente válido, sino también *eficiente*, adaptándose a la heterocedasticidad (volatilidad cambiante) intrínseca de los datos secuenciales.

### 2.9.2 Supuestos Fundamentales del Marco Propuesto

Para transitar de la teoría de Vovk a procesos dependientes, se han identificado los siguientes supuestos como pilares necesarios para el desarrollo de una prueba de consistencia futura:

1. **Estacionariedad y Ergodicidad:** Se asume que el proceso $\{Z_t\}$ es estrictamente estacionario y ergódico. Esto garantiza que los promedios temporales observados en la ventana de datos converjan a los promedios del ensamble, permitiendo que el sistema "aprenda" de la historia pasada.
2. **Condición de $\alpha$-mixing (Mezcla Fuerte):** Para manejar la dependencia, se requiere que el proceso sea $\alpha$-mixing con coeficientes que decaigan algebraicamente ($\alpha(k) \leq Ck^{-\beta}, \beta > 2$). Este supuesto es crucial para aplicar teoremas límite central y asegurar que las observaciones lejanas en el tiempo sean casi independientes.
3. **Regularidad de Lipschitz:** A diferencia del enfoque de histograma de celdas discretas, aquí se asume que tanto la función de regresión $\mu(x)$ como la distribución de los residuos $G(s|x)$ son Lipschitz continuas respecto al espacio de covariables. Esto asegura que puntos cercanos en el tiempo y espacio tengan comportamientos predictivos similares.

### 2.9.3 Mecanismo Propuesto: Transductor Conformal por Kernel

En lugar del enfoque de histogramas anidados de Vovk, este marco propone una arquitectura adaptativa basada en dos componentes:

- **Ventana Temporal Móvil ($L_n$):** Un mecanismo de truncamiento que selecciona las últimas $L_n$ observaciones. Para alcanzar la consistencia, el tamaño de esta ventana debe crecer con $n$ pero a un ritmo controlado ($L_n \to \infty$).
- **Suavizado Espacial por Kernel ($K, h_n$):** En lugar de asignar pesos uniformes a la celda (como en Mondrian), se propone el uso de pesos de relevancia espacial $w_i = K(\frac{d(X_i, X_{n+1})}{h_n})$. Esto permite que el sistema pondere los residuos pasados no solo por su cercanía temporal, sino por su similitud en el espacio de características.

### 2.9.4 Discusión y Perspectivas Futuras

Esta formulación plantea que la convergencia de la integral $\int f dQ_n$ hacia la esperanza condicional real depende del balance entre el sesgo del kernel y la varianza inducida por la dependencia de los datos. Mientras que Vovk utiliza el Teorema de Lévy para martingalas, el análisis en series de tiempo requiere el uso de técnicas de *análisis de sesgo-varianza para estimadores no paramétricos en procesos mixing*.

Es importante notar que este planteamiento se presenta como una *hoja de ruta teórica*. La validación de que este transductor conformal ergódico alcanza la consistencia universal bajo cualquier proceso mixing representaría una extensión significativa del trabajo original de Vovk, unificando la robustez de la predicción conformal con la flexibilidad de la estimación no paramétrica para datos secuenciales de alta complejidad.

# 3 Diseño de la Simulación

Este capítulo describe el diseño experimental desarrollado para evaluar el desempeño de los métodos de pronóstico probabilístico en series temporales. Se presenta la justificación de los escenarios de evaluación, la metodología de simulación empleada y las características específicas de los procesos generadores de datos utilizados.

## 3.1 Introducción

La evaluación rigurosa de metodologías de pronóstico probabilístico requiere un marco experimental controlado que permita comparar el desempeño de diferentes técnicas bajo condiciones conocidas. A diferencia de los estudios con datos reales, donde la distribución verdadera es desconocida y la evaluación se limita a métricas indirectas, los estudios de simulación ofrecen la ventaja fundamental de conocer exactamente el proceso generador de datos (DGP, por sus siglas en inglés) (Rob J Hyndman and Athanasopoulos 2021).

Este conocimiento del DGP permite evaluar directamente la calidad de las distribuciones predictivas mediante su comparación con la verdadera distribución teórica. En particular, el uso del ECRPS (Expected Continuous Ranked Probability Score) como métrica principal de evaluación se justifica porque permite cuantificar simultáneamente la calibración y la nitidez de los pronósticos probabilísticos, comparando las muestras generadas por cada método con muestras de la distribución teórica verdadera (Gneiting and Katzfuss 2014).

El diseño experimental desarrollado considera tres dimensiones fundamentales de variación: (1) la estructura temporal del proceso (estacionariedad y linealidad), (2) la distribución del término de error, y (3) la magnitud de la varianza del ruido. Esta combinación genera un espacio de escenarios suficientemente amplio para evaluar la robustez y adaptabilidad de los métodos bajo diferentes condiciones operativas.

## 3.2 Diseño de la Simulación

### 3.2.1 Selección de Escenarios de Evaluación

El presente estudio considera tres escenarios fundamentales que caracterizan diferentes clases de comportamiento en series temporales. La selección de estos escenarios se fundamenta en la clasificación teórica de procesos estocásticos y en consideraciones de relevancia práctica.

#### Escenario 1: Lineal Estacionario (ARMA)

El primer escenario considera procesos autorregresivos de media móvil (ARMA), que representan la clase fundamental de modelos lineales estacionarios. Un proceso ARMA$(p, q)$ se caracteriza por su capacidad de capturar tanto la persistencia temporal (componente AR) como la dependencia de shocks pasados (componente MA), manteniendo propiedades estadísticas constantes en el tiempo (Arrieta Prieto 2017).

La estacionariedad de estos procesos garantiza que la media, varianza y estructura de autocorrelación permanezcan invariantes bajo traslaciones temporales, lo que facilita la modelación y el pronóstico (Rob J Hyndman and Athanasopoulos 2021). Este escenario permite evaluar el desempeño de los métodos en condiciones ideales, donde los supuestos fundamentales de muchas técnicas estadísticas se cumplen.

#### Escenario 2: Lineal No Estacionario (ARIMA)

El segundo escenario aborda procesos autorregresivos integrados de media móvil (ARIMA), que extienden la clase ARMA para series con tendencias estocásticas. La presencia de raíces unitarias en el polinomio autorregresivo genera comportamientos de paseo aleatorio que son comunes en series económicas y financieras (Rob J Hyndman and Athanasopoulos 2021).

La no estacionariedad introduce desafíos adicionales para el pronóstico probabilístico, ya que la incertidumbre crece sin límite conforme aumenta el horizonte de predicción. Este escenario permite evaluar la capacidad de los métodos para adaptarse a estructuras no estacionarias mediante diferenciación o técnicas adaptativas.

### Escenario 3: No Lineal Estacionario (SETAR)

El tercer escenario considera modelos autorregresivos de umbral auto-excitados (SETAR), que permiten cambios estructurales endógenos en la dinámica del proceso. Estos modelos capturan no linealidades mediante el cambio de régimen determinado por valores pasados de la propia serie (P. Chen and Semmler 2023).

La estacionariedad global de un proceso SETAR requiere condiciones específicas sobre los parámetros autorregresivos en cada régimen y la frecuencia de transición entre regímenes. Estas condiciones se discuten en detalle en la Sección 3.3.3. Este escenario es particularmente relevante para evaluar la capacidad de los métodos conformales de capturar dinámicas asimétricas y dependientes del estado del sistema.

### Ausencia del Escenario No Lineal No Estacionario

La combinación de no linealidad y no estacionariedad, aunque teóricamente posible, presenta desafíos metodológicos sustanciales que la excluyen del alcance de este estudio. Los modelos que combinan ambas características (por ejemplo, SETAR con raíces unitarias condicionales o modelos de cambio de régimen con deriva) requieren condiciones de estabilidad extremadamente restrictivas y su caracterización teórica es un área de investigación activa (P. Chen and Semmler 2023).

Más fundamentalmente, la validez teórica de muchos métodos de predicción conformal, incluyendo aquellos basados en el enfoque de Barber et al. (Barber et al. 2023), asume que el proceso subyacente es al menos localmente estacionario o que las desviaciones de la estacionariedad son graduales y pueden ser capturadas mediante esquemas de ponderación adaptativos. La presencia simultánea de cambios estructurales abruptos (no linealidad) y tendencias estocásticas persistentes (no estacionariedad) violaría estos supuestos fundamentales, invalidando las garantías teóricas de cobertura.

Por estas razones, el presente estudio se enfoca en los tres escenarios anteriores, que permiten una evaluación rigurosa y teóricamente fundamentada del desempeño de los métodos.

### 3.2.2 Estructura del Diseño Factorial

El diseño experimental implementa un esquema factorial completo que combina sistemáticamente tres dimensiones de variación para cada uno de los tres escenarios considerados. Esta estructura genera un total de 420 configuraciones únicas de simulación, distribuidas equitativamente entre los escenarios.

#### Dimensión 1: Configuraciones Paramétricas del Proceso

Para cada clase de modelo (ARMA, ARIMA, SETAR), se consideran 7 configuraciones paramétricas distintas que representan diferentes grados de complejidad y características dinámicas. Las especificaciones detalladas de estas configuraciones se presentan en la Sección 3.3. Esta diversidad paramétrica permite evaluar la sensibilidad de los métodos a diferentes estructuras de dependencia temporal.

#### Dimensión 2: Distribuciones del Término de Error

Se consideran cinco familias de distribuciones para el término de innovación $\varepsilon_t$, seleccionadas para representar diferentes características de forma, simetría y comportamiento en las colas:

1. **Normal:** $\varepsilon_t \sim N(0, \sigma^2)$. Representa el caso base con colas ligeras y simetría perfecta.
2. **T-Student:** $\varepsilon_t \sim \sigma \cdot \frac{t_{18}}{\sqrt{18/16}}$, donde $t_{18}$ denota una distribución t de Student con 18 grados de libertad. Esta parametrización garantiza varianza unitaria y genera colas más pesadas que la normal, capturando eventos extremos más frecuentes.
3. **Exponencial:** $\varepsilon_t \sim \sigma(Y - 1)$, donde $Y \sim \text{Exp}(1)$. Produce asimetría positiva y es relevante para series que modelan variables intrínsecamente positivas o con shocks unidireccionales.
4. **Uniforme:** $\varepsilon_t \sim U(-\sqrt{3}\sigma, \sqrt{3}\sigma)$. Genera soporte acotado y ausencia de colas, representando un caso extremo de curtosis negativa.
5. **Mixtura de Normales:** $\varepsilon_t \sim 0.75 \cdot N(-\sigma/4, \sigma^2/16) + 0.25 \cdot N(3\sigma/4, \sigma^2/16)$. Produce bimodalidad y permite evaluar el desempeño bajo distribuciones predictivas complejas con múltiples modas.

Esta selección permite evaluar la robustez de los métodos ante desviaciones del supuesto de normalidad que frecuentemente se asume en la literatura de pronóstico (Arrieta Prieto 2017).

#### Dimensión 3: Niveles de Varianza del Error

Se consideran cuatro niveles de varianza $\sigma^2 \in \{0.2, 0.5, 1.0, 3.0\}$ que representan diferentes razones señal-ruido. El nivel base $\sigma^2 = 1.0$ corresponde a la parametrización estándar, mientras que $\sigma^2 = 0.2$ representa un escenario de alta predictibilidad y $\sigma^2 = 3.0$ captura situaciones de alta volatilidad donde la incertidumbre inherente domina la dinámica del sistema.

#### Combinatoria Total

La combinación factorial de estas tres dimensiones genera:

$$N_{\text{config}} = 7 \text{ modelos} \times 5 \text{ distribuciones} \times 4 \text{ varianzas} = 140 \text{ configuraciones por escenario} \tag{3-1}$$

Con tres escenarios (ARMA, ARIMA, SETAR), el espacio experimental completo comprende:

$$N_{\text{total}} = 140 \times 3 = 420 \text{ configuraciones únicas} \tag{3-2}$$

Adicionalmente, considerando que cada configuración se evalúa en un horizonte de predicción de 12 pasos usando ventana rodante para que se realice predicción a un paso adelante, el número total de combinaciones configuración-horizonte es de $420 \times 12 = 5040$.

### 3.2.3 Protocolo de Simulación y Partición de Datos

Para cada una de las 420 configuraciones, se implementa el siguiente protocolo de simulación:

1. **Generación de la Serie:** Se simulan $n_{\text{total}} = 302$ observaciones del proceso especificado, precedidas por un período de burn-in de 50 observaciones que se descartan para eliminar el efecto de las condiciones iniciales. Esto resulta en una serie efectiva

de longitud $n = 252$.

2. **Partición Tripartita:** La serie se divide en tres conjuntos disjuntos:
   - **Conjunto de Entrenamiento:** $n_{\text{train}} = 200$ observaciones iniciales utilizadas para la estimación inicial de parámetros y el ajuste de hiperparámetros.
   - **Conjunto de Calibración:** $n_{\text{cal}} = 40$ observaciones subsecuentes utilizadas para la calibración de intervalos de predicción y la construcción de distribuciones conformales.
   - **Conjunto de Prueba:** $n_{\text{test}} = 12$ observaciones finales utilizadas para la evaluación del desempeño predictivo.
3. **Esquema de Ventana Rodante:** La evaluación se realiza mediante una ventana rodante (rolling window) donde:
   - Para el primer paso de predicción, se utilizan las primeras 200 observaciones para entrenamiento y las siguientes 40 para calibración.
   - Para cada paso $h = 1, \ldots, 12$, la ventana de entrenamiento se extiende para incluir las observaciones anteriores, manteniendo fijo el conjunto de calibración de tamaño 40 inmediatamente anterior al punto de predicción.
   - Este esquema emula una situación operativa donde el analista actualiza periódicamente los modelos conforme nueva información se hace disponible.
4. **Generación de Distribuciones Predictivas:** Para cada método y cada paso de predicción $h$, se generan muestras de la distribución predictiva. Estas muestras se comparan con muestras de la distribución teórica verdadera del proceso (conocida por construcción del DGP) mediante el cálculo del ECRPS para ese paso específico.

Este protocolo garantiza que la evaluación sea tanto rigurosa (mediante la comparación con la distribución verdadera) como realista (mediante el esquema de ventana rodante que refleja la práctica operativa).

## 3.3 Procesos Generadores de Datos

Esta sección describe formalmente los modelos utilizados como procesos generadores de datos en cada escenario, junto con las configuraciones paramétricas específicas consideradas. Para cada clase de modelo, se presentan las ecuaciones fundamentales, las condiciones de estacionariedad (cuando corresponda) y las parametrizaciones concretas evaluadas.

### 3.3.1 Procesos ARMA: Escenario Lineal Estacionario

#### Definición y Representación

Un proceso autorregresivo de media móvil de órdenes $p$ y $q$, denotado ARMA$(p, q)$, se define mediante la ecuación en diferencias estocástica:

$$Y_t = c + \sum_{i=1}^{p} \phi_i Y_{t-i} + \varepsilon_t + \sum_{j=1}^{q} \theta_j \varepsilon_{t-j} \tag{3-3}$$

donde $c$ es un término constante, $\{\phi_i\}_{i=1}^{p}$ son los coeficientes autorregresivos, $\{\theta_j\}_{j=1}^{q}$ son los coeficientes de media móvil, y $\{\varepsilon_t\}$ es un proceso de ruido blanco con media cero y varianza $\sigma^2$.

Utilizando el operador de rezagos $L$ definido por $L^k Y_t = Y_{t-k}$, el proceso puede expresarse en forma compacta:

$$\Phi(L)Y_t = c + \Theta(L)\varepsilon_t \tag{3-4}$$

donde $\Phi(L) = 1 - \sum_{i=1}^{p} \phi_i L^i$ es el polinomio autorregresivo y $\Theta(L) = 1 + \sum_{j=1}^{q} \theta_j L^j$ es el polinomio de media móvil.

#### Condiciones de Estacionariedad e Invertibilidad

La estacionariedad y la invertibilidad de un proceso ARMA están determinadas por las raíces de sus polinomios característicos (Rob J Hyndman and Athanasopoulos 2021):

- **Estacionariedad:** El proceso es estacionario en covarianza si y solo si todas las raíces del polinomio autorregresivo $\Phi(z) = 0$ se encuentran estrictamente fuera del

círculo unitario complejo. Equivalentemente, las raíces del polinomio $\Phi(L)$ deben satisfacer $|z_i| > 1$ para todo $i$.

- **Invertibilidad:** El proceso es invertible si y solo si todas las raíces del polinomio de media móvil $\Theta(z) = 0$ se encuentran estrictamente fuera del círculo unitario complejo.

Estas condiciones garantizan que el proceso admite representaciones de Wold (MA($\infty$)) y autorregresiva (AR($\infty$)) convergentes, lo que es fundamental para la teoría de pronóstico (Arrieta Prieto 2017).

### Distribución Predictiva Verdadera

Para un proceso ARMA estacionario e invertible, la distribución del siguiente valor $Y_{n+1}$ condicionada a la historia observada $\mathcal{F}_n = \{Y_1, \ldots, Y_n, \varepsilon_1, \ldots, \varepsilon_n\}$ tiene una forma analítica explícita. Dado que el modelo es lineal, la distribución condicional está completamente caracterizada por su media y varianza condicionales.

La media condicional se obtiene de la ecuación estructural del modelo:

$$\mathbb{E}[Y_{n+1} \mid \mathcal{F}_n] = c + \sum_{i=1}^{p} \phi_i Y_{n+1-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{n+1-j} \tag{3-5}$$

donde todos los términos del lado derecho son conocidos. La varianza condicional es constante e igual a la varianza del ruido:

$$\mathrm{Var}[Y_{n+1} \mid \mathcal{F}_n] = \sigma^2 \tag{3-6}$$

Por lo tanto, si el ruido $\varepsilon_t$ sigue una distribución $F$ con media cero y varianza $\sigma^2$, la distribución predictiva verdadera es:

$$Y_{n+1} \mid \mathcal{F}_n \sim F\left(\mathbb{E}[Y_{n+1} \mid \mathcal{F}_n], \sigma^2\right) \tag{3-7}$$

Esta distribución puede evaluarse numéricamente generando una muestra grande de errores

futuros $\varepsilon_{n+1}^{(b)} \sim F(0, \sigma^2)$ y calculando:

$$Y_{n+1}^{(b)} = c + \sum_{i=1}^{p} \phi_i Y_{n+1-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{n+1-j} + \varepsilon_{n+1}^{(b)}, \quad b = 1, \ldots, B \tag{3-8}$$

donde $B$ es un número suficientemente grande (en esta investigación, $B = 1000$). Esta muestra empírica aproxima la distribución predictiva verdadera y sirve como referencia para el cálculo del ECRPS.

### Configuraciones Paramétricas Evaluadas

La Tabla **3-1** presenta las siete configuraciones ARMA consideradas en este estudio. La selección incluye modelos puramente autorregresivos [AR(1), AR(2)], puramente de media móvil [MA(1), MA(2)], y mixtos [ARMA(1,1), ARMA(2,2), ARMA(2,1)], con diferentes grados de persistencia temporal y complejidad estructural.

| **Nombre** | $p$ | $q$ | $\boldsymbol{\phi}$ | $\boldsymbol{\theta}$ |
|---|---|---|---|---|
| AR(1) | 1 | 0 | $[0.9]$ | $[]$ |
| AR(2) | 2 | 0 | $[0.5, -0.3]$ | $[]$ |
| MA(1) | 0 | 1 | $[]$ | $[0.7]$ |
| MA(2) | 0 | 2 | $[]$ | $[0.4, 0.2]$ |
| ARMA(1,1) | 1 | 1 | $[0.6]$ | $[0.3]$ |
| ARMA(2,2) | 2 | 2 | $[0.4, -0.2]$ | $[0.5, 0.1]$ |
| ARMA(2,1) | 2 | 1 | $[0.7, 0.2]$ | $[0.5]$ |

Table **3-1**: Configuraciones paramétricas para procesos ARMA.

Todas las configuraciones fueron verificadas para satisfacer las condiciones de estacionariedad e invertibilidad mediante el cálculo numérico de las raíces de los polinomios característicos correspondientes.

### 3.3.2 Procesos ARIMA: Escenario Lineal No Estacionario

#### Definición y Operador de Diferenciación

Un proceso autorregresivo integrado de media móvil de órdenes $(p, d, q)$, denotado ARIMA$(p, d, q)$, se construye aplicando el operador de diferenciación $\Delta = 1 - L$ un total de $d$ veces a una serie $Y_t$ y modelando la serie diferenciada resultante $W_t = \Delta^d Y_t$ mediante un proceso ARMA$(p, q)$ estacionario:

$$\Phi(L)W_t = c + \Theta(L)\varepsilon_t \tag{3-9}$$

donde $W_t = (1 - L)^d Y_t$.

Equivalentemente, en términos de la serie original:

$$\Phi(L)(1 - L)^d Y_t = c + \Theta(L)\varepsilon_t \tag{3-10}$$

El orden de integración $d$ representa el número de raíces unitarias en el polinomio autorregresivo ampliado. En la gran mayoría de aplicaciones prácticas, $d \in \{0, 1, 2\}$, siendo $d = 1$ el caso más frecuente (Rob J Hyndman and Athanasopoulos 2021).

#### Propiedades de Estacionariedad

Un proceso ARIMA$(p, d, q)$ es no estacionario por construcción cuando $d > 0$, debido a la presencia de raíces unitarias. Sin embargo, la serie diferenciada $W_t = \Delta^d Y_t$ es estacionaria si el componente ARMA$(p, q)$ subyacente satisface las condiciones de estacionariedad e invertibilidad descritas en la Sección 3.3.1.

Esta propiedad de *estacionariedad en diferencias* es fundamental para el pronóstico, ya que permite aplicar toda la teoría desarrollada para procesos estacionarios a la serie transformada $W_t$, recuperando posteriormente los pronósticos en la escala original mediante integración sucesiva (Rob J Hyndman and Athanasopoulos 2021).

#### Distribución Predictiva Verdadera

Para un proceso ARIMA$(p, d, q)$, la distribución del siguiente valor $Y_{n+1}$ condicionada a la historia observada se obtiene mediante un procedimiento de dos etapas que explota la

estructura de diferenciación del modelo.

Primero, se predice el siguiente valor de la serie diferenciada $W_{n+1} = \Delta^d Y_{n+1}$ usando la distribución ARMA subyacente. Para el caso más común $d = 1$, la serie diferenciada es:

$$W_t = Y_t - Y_{t-1} \tag{3-11}$$

y su predicción un paso adelante, condicionada a la historia $\mathcal{F}_n = \{Y_1, \ldots, Y_n, \varepsilon_1, \ldots, \varepsilon_n\}$, sigue la distribución ARMA:

$$W_{n+1} \mid \mathcal{F}_n \sim F\left(\mathbb{E}[W_{n+1} \mid \mathcal{F}_n], \sigma^2\right) \tag{3-12}$$

donde:

$$\mathbb{E}[W_{n+1} \mid \mathcal{F}_n] = c + \sum_{i=1}^{p} \phi_i W_{n+1-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{n+1-j} \tag{3-13}$$

Segundo, se recupera la predicción en la escala original mediante la relación de integración:

$$Y_{n+1} = Y_n + W_{n+1} \tag{3-14}$$

Por lo tanto, la distribución predictiva verdadera para $Y_{n+1}$ es:

$$Y_{n+1} \mid \mathcal{F}_n \sim F\left(Y_n + \mathbb{E}[W_{n+1} \mid \mathcal{F}_n], \sigma^2\right) \tag{3-15}$$

Esta distribución puede evaluarse numéricamente generando muestras del incremento futuro:

$$W_{n+1}^{(b)} = c + \sum_{i=1}^{p} \phi_i W_{n+1-i} + \sum_{j=1}^{q} \theta_j \varepsilon_{n+1-j} + \varepsilon_{n+1}^{(b)} \tag{3-16}$$

y aplicando la transformación:

$$Y_{n+1}^{(b)} = Y_n + W_{n+1}^{(b)}, \quad b = 1, \ldots, B \tag{3-17}$$

donde $\varepsilon_{n+1}^{(b)} \sim F(0, \sigma^2)$ son errores futuros independientes. Esta muestra empírica representa la distribución predictiva verdadera que sirve como referencia para el ECRPS.

#### Configuraciones Paramétricas Evaluadas

La Tabla **3-2** presenta las siete configuraciones ARIMA$(p, 1, q)$ consideradas en este estudio. Todas las configuraciones utilizan $d = 1$, reflejando el caso más común en aplicaciones económicas y financieras. La selección incluye desde el paseo aleatorio puro [ARIMA(0,1,0)] hasta modelos con estructura autorregresiva y de media móvil en la serie diferenciada.

| **Nombre** | $p$ | $d$ | $q$ | $\phi$ | $\theta$ |
|---|---|---|---|---|---|
| ARIMA(0,1,0) | 0 | 1 | 0 | [] | [] |
| ARIMA(1,1,0) | 1 | 1 | 0 | [0.6] | [] |
| ARIMA(2,1,0) | 2 | 1 | 0 | $[0.5, -0.2]$ | [] |
| ARIMA(0,1,1) | 0 | 1 | 1 | [] | [0.5] |
| ARIMA(0,1,2) | 0 | 1 | 2 | [] | $[0.4, 0.25]$ |
| ARIMA(1,1,1) | 1 | 1 | 1 | [0.7] | $[-0.3]$ |
| ARIMA(2,1,2) | 2 | 1 | 2 | $[0.6, 0.2]$ | $[0.4, -0.1]$ |

Table **3-2**: Configuraciones paramétricas para procesos ARIMA.

### 3.3.3 Procesos SETAR: Escenario No Lineal Estacionario

#### Definición y Mecanismo de Cambio de Régimen

Un modelo autorregresivo de umbral auto-excitado con dos regímenes, denotado SETAR(2; $p_1$, $p_2$), se define mediante una estructura de cambio de régimen determinado por valores pasados de la propia serie (P. Chen and Semmler 2023):

$$Y_t = \begin{cases} \phi_0^{(1)} + \sum_{i=1}^{p_1} \phi_i^{(1)} Y_{t-i} + \varepsilon_t^{(1)} & \text{si } Y_{t-d} \leq r \\ \phi_0^{(2)} + \sum_{i=1}^{p_2} \phi_i^{(2)} Y_{t-i} + \varepsilon_t^{(2)} & \text{si } Y_{t-d} > r \end{cases} \tag{3-18}$$

donde:

- $r$ es el *valor umbral* (threshold value) que determina el cambio de régimen
- $d$ es el *rezago de umbral* (threshold delay) que especifica qué valor pasado de la serie se utiliza para determinar el régimen activo

- $\phi_0^{(j)}$ y $\{\phi_i^{(j)}\}_{i=1}^{p_j}$ son los parámetros específicos del régimen $j$
- $\varepsilon_t^{(j)} \sim WN(0, \sigma_j^2)$ son procesos de ruido blanco que pueden tener varianzas diferentes en cada régimen

La notación SETAR(2; $d$, $p$) denota un modelo de dos regímenes con rezago de umbral $d$ y orden autorregresivo común $p$ en ambos regímenes (aunque en general $p_1$ y $p_2$ pueden diferir).

### Estacionariedad en Procesos SETAR

La estacionariedad de procesos SETAR es sustancialmente más compleja que en modelos lineales, ya que la dinámica cambia endógenamente según el estado del sistema. Las condiciones suficientes para la estacionariedad han sido objeto de extensa investigación (P. Chen and Semmler 2023).

**Caso SETAR(2; 1, 1):** Para el caso más simple de dos regímenes con orden autorregresivo 1, Petruccelli and Woolford 1984 demostraron que el proceso es ergódico si y solo si:

$$|\phi_1^{(1)}| < 1, \quad |\phi_1^{(2)}| < 1, \quad \text{y} \quad |\phi_1^{(1)}\phi_1^{(2)}| < 1 \tag{3-19}$$

Esta condición requiere que cada régimen sea individualmente estable y que el producto de los coeficientes autorregresivos sea menor que uno en valor absoluto. Esta última condición captura el efecto de la interacción entre regímenes.

**Caso General SETAR(2; $p_1$, $p_2$):** Para órdenes autorregresivos mayores,Chan and Tong 1985 proporcionaron una condición suficiente basada en el radio espectral de las matrices compañeras:

$$\max_j \sum_{i=1}^{p_j} |\phi_i^{(j)}| < 1 \tag{3-20}$$

Sin embargo, esta condición es bastante conservadora. Un criterio más general y menos restrictivo se basa en el concepto de *radio espectral conjunto* (joint spectral radius) de las matrices compañeras de ambos regímenes (P. Chen and Semmler 2023). Sea $\mathbf{\Phi}^{(j)}$ la matriz compañera del régimen $j$:

$$\mathbf{\Phi}^{(j)} = \begin{pmatrix} \phi_1^{(j)} & \phi_2^{(j)} & \cdots & \phi_{p-1}^{(j)} & \phi_p^{(j)} \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \end{pmatrix} \tag{3-21}$$

El radio espectral conjunto se define como:

$$\rho(\{\mathbf{\Phi}^{(1)}, \mathbf{\Phi}^{(2)}\}) = \lim_{k\to\infty} \max \|\mathbf{\Phi}^{(i_1)} \cdots \mathbf{\Phi}^{(i_k)}\|^{1/k} \tag{3-22}$$

donde el máximo se toma sobre todas las secuencias posibles de $k$ matrices.

El proceso SETAR es estacionario si $\rho(\{\mathbf{\Phi}^{(1)}, \mathbf{\Phi}^{(2)}\}) < 1$. Este criterio es menos restrictivo que (3-20) y permite que algunos regímenes individuales sean incluso explosivos, siempre que la dinámica global del sistema sea estabilizadora (P. Chen and Semmler 2023).

### Distribución Predictiva Verdadera

La distribución del siguiente valor $Y_{n+1}$ en un proceso SETAR condicionada a la historia observada $\mathcal{F}_n = \{Y_1, \ldots, Y_n, \varepsilon_1, \ldots, \varepsilon_n\}$ depende críticamente del régimen que será activado en el tiempo $n+1$. A diferencia de los modelos lineales, la predicción requiere determinar primero qué régimen gobernará la dinámica futura.

El régimen activo en el tiempo $n+1$ se determina comparando el valor retardado $Y_{n+1-d}$ con el umbral $r$:

$$\text{Régimen}_{n+1} = \begin{cases} 1 & \text{si } Y_{n+1-d} \le r \\ 2 & \text{si } Y_{n+1-d} > r \end{cases} \tag{3-23}$$

Dado que $Y_{n+1-d}$ ya es conocido en el tiempo $n$ (pues $n+1-d \le n$ para $d \ge 1$), el régimen futuro es determinístico y no hay incertidumbre sobre cuál dinámica aplicar. Una vez identificado el régimen $j \in \{1, 2\}$, la media condicional se calcula mediante:

$$\mathbb{E}[Y_{n+1} \mid \mathcal{F}_n, \text{Régimen}_{n+1} = j] = \phi_0^{(j)} + \sum_{i=1}^{p_j} \phi_i^{(j)} Y_{n+1-i} \tag{3-24}$$

donde todos los valores $Y_{n+1-i}$ en el lado derecho son observados. La varianza condicional

es constante dentro de cada régimen:

$$\text{Var}[Y_{n+1} \mid \mathcal{F}_n, \text{Régimen}_{n+1} = j] = \sigma_j^2 \tag{3-25}$$

Por lo tanto, la distribución predictiva verdadera es:

$$Y_{n+1} \mid \mathcal{F}_n \sim F_j \left( \mathbb{E}[Y_{n+1} \mid \mathcal{F}_n, \text{Régimen}_{n+1} = j], \sigma_j^2 \right) \tag{3-26}$$

donde $F_j$ es la distribución del ruido en el régimen $j$ y el subíndice $j$ se determina mediante (3-23).

Esta distribución puede evaluarse numéricamente generando una muestra grande de errores futuros específicos del régimen activo:

$$Y_{n+1}^{(b)} = \phi_0^{(j)} + \sum_{i=1}^{p_j} \phi_i^{(j)} Y_{n+1-i} + \varepsilon_{n+1}^{(b)}, \quad b = 1, \ldots, B \tag{3-27}$$

donde $\varepsilon_{n+1}^{(b)} \sim F_j(0, \sigma_j^2)$ son errores independientes del régimen determinado. A diferencia de los modelos ARMA, aquí no existe incertidumbre sobre el régimen en predicciones un paso adelante, lo que simplifica considerablemente la evaluación de la distribución predictiva verdadera.

### Configuraciones Paramétricas Evaluadas

La Tabla **3-3** presenta las siete configuraciones SETAR consideradas en este estudio. Las configuraciones incluyen diferentes órdenes autorregresivos, rezagos de umbral y valores de umbral, representando una amplia gama de comportamientos no lineales.

Finalmente, es importante destacar que todas las configuraciones detalladas en la Tabla **3-3** fueron seleccionadas bajo un estricto criterio de estabilidad. Para garantizar el rigor estadístico de las comparaciones en este escenario, se realizó un análisis de estacionariedad basado en el cálculo numérico del radio espectral conjunto ($\rho$) para cada par de matrices compañeras. Se verificó que en la totalidad de los casos empleados en la simulación se cumple la condición $\rho < 1$, asegurando que los procesos SETAR generados son globalmente estacionarios.

| Nombre | $\phi^{(1)}$ | $\phi^{(2)}$ | $r$ | $d$ |
|---|---|---|---|---|
| SETAR-1 | [0.6] | [-0.5] | 0.0 | 1 |
| SETAR-2 | [0.7] | [-0.7] | 0.0 | 2 |
| SETAR-3 | [0.5, -0.2] | [-0.3, 0.1] | 0.5 | 1 |
| SETAR-4 | [0.8, -0.15] | [-0.6, 0.2] | 1.0 | 2 |
| SETAR-5 | [0.4, -0.1, 0.05] | [-0.3, 0.1, -0.05] | 0.0 | 1 |
| SETAR-6 | [0.5, -0.3, 0.1] | [-0.4, 0.2, -0.05] | 0.5 | 2 |
| SETAR-7 | [0.3, 0.1] | [-0.2, -0.1] | 0.8 | 3 |

Table **3-3**: Configuraciones paramétricas para procesos SETAR.

## 3.4 Modelos predictivos

Para evaluar la capacidad de cuantificación de la incertidumbre en diversos entornos estocásticos, esta investigación emplea un conjunto heterogéneo de nueve modelos predictivos. Esta selección abarca desde métodos de remuestreo clásicos y propuestas de predicción conformal, hasta arquitecturas de aprendizaje profundo y modelos híbridos de diseño propio. El uso de esta diversidad de enfoques permite contrastar cómo las garantías teóricas de cada familia de modelos se traducen en un rendimiento práctico bajo la métrica ECRPS, especialmente cuando se enfrentan a la ruptura de los supuestos de intercambiabilidad y linealidad.

### 3.4.1 Circular Block Bootstrap (CBB)

#### Propuesta Teórica

El método *Circular Block Bootstrap*, introducido originalmente por Politis and Romano (1992), surge como una evolución técnica para corregir las deficiencias del remuestreo por bloques convencional. Según detalla Lahiri (2003), el problema fundamental de los métodos de bloques no circulares (como el MBB) es que las observaciones situadas en los extremos de la serie temporal aparecen con menos frecuencia en los bloques resampleados, lo que genera una infra-representación de los bordes y un sesgo en la estimación de la varianza.

Teóricamente, el CBB soluciona esto mediante la "circunscripción" de los datos: se asume que la serie temporal $\{X_1, \ldots, X_n\}$ se encuentra sobre un círculo, de modo que el dato

$X_n$ es seguido inmediatamente por $X_1$. Esta extensión periódica permite que el algoritmo defina $n$ bloques posibles de longitud $l$, garantizando que cada valor histórico tenga una probabilidad idéntica $(1/n)$ de ser seleccionado. Esta propiedad de equiprobabilidad es crucial para obtener distribuciones predictivas mejor calibradas y estadísticamente consistentes bajo dependencia temporal (Lahiri 2003).

### De la Teoría a la Práctica

En la implementación desarrollada para este estudio, se realizaron adaptaciones específicas para integrar el modelo en un flujo de trabajo de pronóstico probabilístico iterativo. La teoría indica que el rendimiento del bootstrap depende críticamente de la longitud del bloque $l$. Para automatizar este proceso sin intervención manual, se integró la heurística propuesta por Politis and White (2004), la cual establece que para datos dependientes, la longitud óptima del bloque puede aproximarse mediante la relación $l \approx 1.5 \times n^{1/3}$.

### Parámetros e Hiperparámetros en la Implementación

El modelo de Circular Block Bootstrap emplea los siguientes parámetros en su configuración:

- **block_length ($l$):** Constituye el hiperparámetro de mayor relevancia en el rendimiento del modelo, ya que determina cuánta dependencia temporal se preserva en las muestras bootstrap. La metodología experimental ofrece cuatro estrategias de configuración:

  1. *Valor fijo manual:* El investigador puede especificar directamente un entero positivo $l \geq 2$ basado en conocimiento experto del proceso.

  2. *Heurística automática:* Se emplea la regla de Politis and White (2004), donde $l \approx 1.5 \times n^{1/3}$, restringiendo el resultado al rango $[2, 50]$ para mantener eficiencia computacional y evitar sobreajuste.

  3. *Optimización mediante búsqueda en rejilla:* Durante la fase de validación, se evalúan cuatro configuraciones específicas basadas en el tamaño del conjunto de entrenamiento ($n_{\text{train}} = 200$):

     - $l = 5$ (bloques pequeños, capturan dependencias de corto plazo)

- $l = 9$ (aproximación de la heurística $1.5 \times 200^{1/3} \approx 8.75$)
- $l = \lfloor \sqrt{n_{\text{train}}} \rfloor = 14$ (bloques medianos, balance entre dependencia y varianza)
- $l = \lfloor n_{\text{train}}/5 \rfloor = 40$ (bloques grandes, preservan estructura temporal extendida)

Esta rejilla fue diseñada para explorar diferentes grados de dependencia temporal, desde bloques que capturan correlaciones inmediatas hasta aquellos que preservan patrones de más largo plazo.

4. *Congelamiento post-optimización:* Una vez seleccionado el $l$ óptimo mediante validación cruzada basada en el CRPS promedio, este valor se mantiene fijo para todos los pasos de predicción rolling en la fase de evaluación. Este congelamiento es crítico para evitar el *data leakage* que resultaría de re-optimizar el hiperparámetro en cada ventana temporal, lo cual introduciría información futura inadmisible en el proceso de selección.

La selección automática del hiperparámetro $l$ mediante la grilla mencionada anteriormente, constituye un elemento crítico del diseño experimental. Esta estrategia permite comparar modelos bajo condiciones equitativas, donde cada método ha sido optimizado con acceso únicamente a información pasada (conjunto de entrenamiento + validación), sin introducir sesgos de optimización que contaminarían la estimación del rendimiento predictivo verdadero en datos futuros (conjunto de prueba).

### 3.4.2 Sieve Bootstrap (SB)

#### Propuesta Teórica

A diferencia de los métodos de remuestreo por bloques, el *Sieve Bootstrap* (o bootstrap de tamiz), introducido formalmente por Bühlmann (1997) y analizado por Lahiri (2003), no intenta preservar la dependencia temporal mediante la partición física de la serie. En su lugar, utiliza una aproximación paramétrica para filtrar la estructura de dependencia, bajo la premisa de que cualquier proceso lineal estacionario admite una representación autorregresiva de orden infinito, AR($\infty$).

Teóricamente, el método consiste en ajustar un modelo autorregresivo de orden finito $p$,

donde $p$ crece con el tamaño de la muestra $n$, de modo que el tamiz autorregresivo capture la estructura de autocorrelación. Una vez ajustado el modelo, se obtienen los residuos, los cuales deben ser idealmente independientes e idénticamente distribuidos (i.i.d.). La distribución predictiva se genera entonces aplicando el bootstrap i.i.d. convencional sobre estos residuos y proyectándolos a través de los coeficientes autorregresivos estimados (Lahiri 2003).

### De la Teoría a la Práctica

En la implementación de este estudio, el Sieve Bootstrap se ha adaptado para operar en un entorno de pronóstico probabilístico iterativo mediante las siguientes consideraciones:

- **Selección de Orden mediante Validación:** Aunque la teoría sugiere que el orden $p$ debe tender a infinito, en la práctica se implementó un mecanismo de selección basado en la evaluación del desempeño predictivo. Durante la fase de optimización, se evalúan tres configuraciones de orden fijo ($p \in \{5, 10, 20\}$) y se selecciona aquella que minimiza el ECRPS (Continuous Ranked Probability Score) sobre el conjunto de validación.
- **Ajuste Durante Calibración:** Una vez identificado el orden óptimo, el modelo autorregresivo se ajusta durante la fase de calibración utilizando todos los datos disponibles en ese momento (entrenamiento + calibración). Este ajuste determina tanto los coeficientes autorregresivos $\hat{\phi}$ como el conjunto de residuos $\hat{\epsilon}$, los cuales se mantienen constantes durante toda la fase de evaluación posterior.
- **Mecanismo de Inferencia Secuencial:** Para generar la predicción probabilística en el tiempo $t + 1$, el modelo utiliza los últimos $p$ valores observados de la serie hasta ese instante, aplicando los coeficientes autorregresivos previamente estimados y añadiendo un residuo seleccionado aleatoriamente mediante bootstrap con reemplazo. Este enfoque permite que el modelo sea computacionalmente eficiente mientras mantiene la capacidad de adaptarse a la evolución más reciente de la serie.

### Parámetros e Hiperparámetros en la Implementación

La configuración del método Sieve Bootstrap se basa en los siguientes parámetros:

- **order ($p$):** Es el hiperparámetro crítico que define la resolución del tamiz autorre-

gresivo. La metodología experimental evalúa tres configuraciones de orden fijo para explorar diferentes niveles de memoria del proceso:

  - $p = 5$ (dependencias de corto plazo)
  - $p = 10$ (memoria intermedia)
  - $p = 20$ (dependencias extendidas)

  La selección del orden óptimo se realiza mediante validación cruzada, eligiendo aquel que minimiza el ECRPS sobre el conjunto de validación de 40 observaciones.

- **Estabilización de parámetros:** Tras identificar el orden óptimo durante la calibración, se ajusta el modelo autorregresivo con los datos completos de entrenamiento y calibración, fijando los coeficientes $\hat{\phi}$ y el conjunto de residuos $\hat{\epsilon}$. Esto garantiza que la comparación entre modelos sea robusta y evita que el rendimiento predictivo se vea afectado por re-estimaciones ruidosas en muestras cambiantes.

- **n_boot:** Tamaño de la distribución predictiva, fijado en 1000 muestras generadas mediante el remuestreo con reemplazo de los residuos centrados.

- **Residuos centrados:** Para cumplir con el supuesto de media cero del modelo AR, los residuos se centran antes del remuestreo: $\tilde{\epsilon}_i = \hat{\epsilon}_i - \bar{\epsilon}$, asegurando que el bootstrap no introduzca sesgos artificiales en la media de la predicción.

### 3.4.3 Least Squares Prediction Machine (LSPM)

#### Propuesta Teórica

El *Least Squares Prediction Machine* (LSPM), introducido formalmente por Vovk, Gammerman, and Shafer (2022), representa una evolución de la predicción conformal que trasciende la generación de intervalos de confianza para construir distribuciones predictivas completas. A diferencia de los predictores conformales estándar tratados en capítulos previos, el LSPM se define como un Sistema Predictivo Conformal (CPS), cuya salida es una Función de Distribución Predictiva Conformal (CPD). Esta función es estadísticamente válida bajo el supuesto de intercambiabilidad, lo que significa que es capaz de cuantificar la incertidumbre sin requerir suposiciones sobre la distribución paramétrica de los errores.

Teóricamente, el LSPM utiliza el método de mínimos cuadrados ordinarios (OLS) como

algoritmo subyacente. Según detalla Vovk, Gammerman, and Shafer (2022), la variante más robusta es el **LSPM Studentizado**. Esta versión es fundamental para el análisis de regresión, ya que utiliza los elementos diagonales de la matriz de proyección o "matriz hat" ($\bar{H}$) para normalizar los residuos. El uso de residuos studentizados garantiza que la distribución predictiva calculada sea monótonamente creciente (cumpliendo el requisito teórico $R1$), incluso cuando el nuevo objeto de prueba posee un alto apalancamiento (*leverage*).

Para un conjunto de datos aumentado que incluye $n-1$ ejemplos de entrenamiento y un nuevo objeto $x_n$ con una etiqueta hipotética $y$, el sistema calcula una serie de valores críticos $C_i$ que actúan como los puntos de salto de la distribución escalonada:

$$C_i := \frac{A_i}{B_i}, \quad i = 1, \ldots, n-1 \tag{3-28}$$

Donde los componentes $A_i$ y $B_i$ para la versión studentizada se definen, de acuerdo con las ecuaciones 7.15 y 7.16 de Vovk, Gammerman, and Shafer (2022), como:

$$B_i := \sqrt{1 - h_{n,n}} + \frac{h_{i,n}}{\sqrt{1 - h_{i,i}}} \tag{3-29}$$

$$A_i := \frac{\sum_{j=1}^{n-1} h_{j,n} y_j}{\sqrt{1 - h_{n,n}}} + \frac{y_i - \sum_{j=1}^{n-1} h_{i,j} y_j}{\sqrt{1 - h_{i,i}}} \tag{3-30}$$

En estas expresiones, $h_{i,j}$ representa el elemento en la fila $i$ y columna $j$ de la matriz $\bar{H} = \bar{X}(\bar{X}^T\bar{X})^{-1}\bar{X}^T$, calculada sobre la matriz de diseño de todos los objetos disponibles.

### De la Teoría a la Práctica

La implementación del LSPM desarrollada para este estudio adapta el marco general de Vovk a las particularidades del pronóstico de series temporales:

- **Transformación Autorregresiva:** Mientras que la teoría original de Vovk asume objetos $x_i$ como vectores de atributos independientes, en esta investigación los objetos se construyen dinámicamente a partir de los retardos (*lags*) de la propia serie temporal. Esto convierte al LSPM en un modelo autorregresivo conformal de orden $p$.

- **Estabilidad Numérica mediante Pseudoinversa:** Para el cálculo de la matriz $\bar{H}$, el código emplea la pseudoinversa de Moore-Penrose. Esta decisión técnica es crítica en la práctica, dado que las series temporales suelen presentar alta autocorrelación, lo que puede derivar en matrices de diseño casi singulares que harían fallar a la inversión matricial estándar.
- **Filtrado de Casos Singulares:** Siguiendo las recomendaciones teóricas sobre la existencia de residuos, la implementación incorpora un umbral de tolerancia ($10^{-10}$) para evitar divisiones por cero en casos donde el apalancamiento ($h_{i,i}$) sea igual a la unidad, situación que ocurre cuando un dato es tan influyente que el modelo lo ajusta sin error residual.

#### Parámetros e Hiperparámetros en la Implementación

La configuración del modelo LSPM se rige por los siguientes parámetros:

- **version:** Configurado permanentemente como `'studentized'`. Se optó por esta versión ya que, matemáticamente, es la única que garantiza la validez de la distribución predictiva sin requerir que el apalancamiento del nuevo objeto sea inferior a 0.5 (Vovk, Gammerman, and Shafer 2022).
- **n_lags ($p$):** Representa la complejidad del modelo (número de retardos). Su gestión sigue un protocolo estricto:
  1. *Heurística de inicialización:* Por defecto, se calcula como $p = \lfloor n^{1/3} \rfloor$, equilibrando la capacidad de captura de patrones contra el riesgo de sobreajuste en muestras pequeñas.
  2. *Congelamiento (Freezing):* Una vez determinada la estructura óptima durante la optimización, el valor de $p$ se congela. Esto asegura que la matriz de diseño mantenga la misma dimensión durante toda la fase de prueba (*rolling forecast*), evitando fugas de información (*data leakage*).
- **random_state:** Aunque el cálculo de los valores críticos $C_i$ es un proceso determinista bajo la teoría conformal, este parámetro se utiliza para inicializar el generador de números aleatorios (`np.random.default_rng`), asegurando la reproducibilidad total si el modelo requiere generar muestras de la CPD para cálculos posteriores de métricas de error.

### 3.4.4 Least Squares Prediction Machine with Weighted Residuals (LSPMW)

#### Propuesta Teórica

El modelo *Least Squares Prediction Machine with Weighted Residuals* (LSPMW) constituye una evolución del LSPM diseñada específicamente para entornos donde el supuesto de intercambiabilidad (*exchangeability*) es invalidado por la presencia de deriva distributiva (*distribution drift*) o dependencias temporales. Esta variante se fundamenta en los desarrollos teóricos de Barber et al. (2023), quienes proponen el uso de **cuantiles ponderados** para otorgar mayor relevancia a las observaciones más recientes y mitigar el sesgo introducido por datos obsoletos.

Teóricamente, Barber et al. demuestran que, ante una violación de la intercambiabilidad, la pérdida de cobertura (denominada *coverage gap*) de un predictor conformal puede acotarse mediante la distancia de variación total ($d_{TV}$) entre la distribución conjunta de los datos originales y la de los datos tras un intercambio de posiciones. Para habilitar la robustez en series temporales, se introducen pesos normalizados $\tilde{w}_i$ para cada residuo $R_i$, de modo que la función de distribución empírica ponderada se define como:

$$\hat{F}_n(y) = \sum_{i=1}^{n} \tilde{w}_i \cdot \delta_{R_i} \tag{3-31}$$

En contextos de deriva gradual, la propuesta teórica óptima sugiere un esquema de decaimiento geométrico para los pesos:

$$w_i = \rho^{n-i}, \quad \text{con } \rho \in (0, 1) \tag{3-32}$$

Donde el hiperparámetro $\rho$ (parámetro de decaimiento) controla la velocidad a la que el modelo "olvida" el pasado. Un valor de $\rho$ cercano a 1 se aproxima al LSPM estándar, mientras que valores menores concentran la masa de probabilidad en el pasado inmediato, reduciendo el error de cobertura a costa de aumentar la varianza de los intervalos de predicción.

### De la Teoría a la Práctica

La implementación del LSPMW en esta investigación traduce el concepto de cuantiles ponderados de Barber et al. (2023) a un mecanismo de generación de distribuciones sintéticas mediante las siguientes adaptaciones:

- **Mecanismo de Expansión Ponderada:** Mientras la teoría se centra en el cálculo de un cuantil específico, la práctica requiere una distribución completa para evaluar la métrica ECRPS. La implementación utiliza un método de *Weighted Expansion*, donde los valores críticos del sistema ($C_i$) se replican en un vector de tamaño fijo (1000 muestras) proporcionalmente a su peso temporal $w_i$.
- **Ajuste Fino de Replicaciones:** Debido a que el producto $\tilde{w}_i \times 1000$ rara vez resulta en un entero, el algoritmo implementa una lógica de ajuste para garantizar que la suma de las replicaciones sea exactamente igual al tamaño objetivo, incrementando o decrementando la frecuencia de los valores con mayor peso relativo.
- **Optimización Dinámica de la Memoria:** El valor óptimo de $\rho$ no es fijo, sino que depende de la volatilidad intrínseca de la serie. Por ello, se integra con un optimizador externo (*TimeBalancedOptimizer*) que busca el valor de decaimiento que minimiza el CRPS en el conjunto de validación.
- **Protocolo de Congelamiento Predictivo:** Para cumplir con el rigor estadístico y evitar el *data leakage*, una vez que se identifica el $\rho$ óptimo durante la calibración, tanto este valor como el vector de valores críticos calculados hasta ese momento se congelan (*is_frozen*), sirviendo como base inmutable para la inferencia sobre el conjunto de prueba.

### Parámetros e Hiperparámetros en la Implementación

La configuración del modelo LSPMW emplea los siguientes elementos críticos:

- **rho ($\rho$):** Es el hiperparámetro fundamental del modelo. Controla la importancia relativa de la historia temporal.
  - *Rango:* $(0, 1)$. En esta investigación, el optimizador evalúa comúnmente valores en el rango $[0.90, 0.99]$.
  - *Impacto:* Valores bajos de $\rho$ permiten una adaptación rápida a cambios de

régimen (ej. escenarios de *changepoints*), pero pueden generar distribuciones predictivas ruidosas si la ventana efectiva de datos se vuelve demasiado pequeña.

- **Estructura de Residuos Congelados:** A diferencia del modelo base, el LSPMW almacena los artefactos `_frozen_critical_values` y `_frozen_weights`. Esto permite que, durante la fase de evaluación, la distribución predictiva no dependa de re-estimaciones autorregresivas que podrían introducir inestabilidad, sino de la estructura de error validada.
- **n_samples_target (1000):** Define la resolución de la distribución predictiva generada. Este valor asegura una precisión suficiente para la integración numérica requerida por el CRPS.
- **n_lags ($p$):** Heredado de la arquitectura LSPM, define el orden autorregresivo del filtro lineal previo al cálculo de los residuos conformales.

### 3.4.5 Mondrian Conformal Predictive System (MCPS)

#### Propuesta Teórica

El *Mondrian Conformal Predictive System* (MCPS), formalizado por Boström, Johansson, and Löfström (2021), representa una extensión localmente adaptativa del Sistema Predictivo Conformal estándar (SCPS). Mientras que el SCPS asume que los errores de predicción se distribuyen homogéneamente sobre todo el espacio de entrada, el MCPS reconoce que esta suposición es frecuentemente violada en aplicaciones reales, donde la incertidumbre puede variar significativamente según el régimen operativo del modelo.

**Fundamento Teórico: Particionamiento Mondrian** La innovación central del MCPS radica en la estrategia de **particionamiento Mondrian** (Vovk, Gammerman, and Shafer 2005), que divide el conjunto de calibración $\mathcal{D}_c$ en subconjuntos disjuntos basándose en características compartidas de las predicciones. Formalmente, sea $h : \mathcal{X} \to \mathbb{R}$ un modelo de regresión entrenado, y sea $B \in \mathbb{N}$ el número de bins (hiperparámetro). Se define una partición:

$$\mathcal{D}_c = \bigcup_{\kappa=1}^{B} \mathcal{D}_c^{\kappa}, \quad \mathcal{D}_c^{\kappa} \cap \mathcal{D}_c^{\kappa'} = \emptyset \text{ para } \kappa \neq \kappa' \tag{3-33}$$

Donde cada subconjunto $\mathcal{D}_c^\kappa$ agrupa instancias de calibración $(x_j, y_j)$ cuyas predicciones $h(x_j)$ caen dentro del mismo rango de valores. Según Boström, Johansson, and Löfström (2021), esta partición se construye típicamente mediante cuantiles empíricos de las predicciones:

$$\mathcal{D}_c^\kappa = \left\{(x_j, y_j) \in \mathcal{D}_c \,:\, q_{\frac{\kappa-1}{B}} \leq h(x_j) < q_{\frac{\kappa}{B}}\right\} \tag{3-34}$$

donde $q_p$ denota el cuantil $p$ de las predicciones $\{h(x_j)\}_{j=1}^{N_c}$.

La intuición detrás de esta estrategia es que instancias con predicciones similares probablemente compartan patrones de error similares. Por ejemplo, en logística de e-commerce (Ye, Hijazi, and Van Hentenryck 2025), órdenes con tiempos de entrega estimados cortos (entregas locales) exhiben una distribución de errores diferente a aquellas con tiempos estimados largos (envíos internacionales). El particionamiento Mondrian captura esta heterogeneidad de forma automática y no paramétrica.

**Cálculo Localizado de Scores Conformales** A diferencia del SCPS, que calcula scores globales sobre todo $\mathcal{D}_c$, el MCPS opera localmente. Para una nueva instancia de prueba $x$ con predicción puntual $h(x)$, se determina primero su bin correspondiente $\kappa^*$ tal que:

$$\kappa^* = \arg\min_\kappa \left\{\kappa \,:\, h(x) < q_{\frac{\kappa}{B}}\right\} \tag{3-35}$$

Los **calibration scores** se calculan entonces únicamente sobre el subconjunto local $\mathcal{D}_c^{\kappa^*} = \{(x_j, y_j)\}_{j=1}^{N_c^{\kappa^*}}$:

$$C_j^{\kappa^*} = h(x) + (y_j - h(x_j)), \quad \forall (x_j, y_j) \in \mathcal{D}_c^{\kappa^*} \tag{3-36}$$

Esta formulación es idéntica a la del SCPS en su estructura algebraica, pero la diferencia crítica reside en que los residuos históricos $(y_j - h(x_j))$ provienen exclusivamente de casos donde el modelo exhibió un comportamiento predictivo similar. Matemáticamente, esto implica que la distribución condicional de errores $P(\epsilon \mid h(x) \in \text{Bin}_\kappa)$ se estima localmente, en lugar de globalmente como en SCPS donde se asume $P(\epsilon \mid h(x)) = P(\epsilon)$.

**Construcción de la Distribución Predictiva** Según Vovk, Gammerman, and Shafer (2022), la función de distribución acumulada (CDF) estimada se construye a partir de

los scores ordenados $C^{\kappa^*}_{(1)} < C^{\kappa^*}_{(2)} < \cdots < C^{\kappa^*}_{(N^{\kappa^*}_c)}$, con fronteras definidas como $C^{\kappa^*}_{(0)} = -\infty$ y $C^{\kappa^*}_{(N^{\kappa^*}_c+1)} = \infty$. Introduciendo una variable aleatoria de suavizado $\tau \sim \text{Uniform}(0,1)$, la CDF localizada se define como:

$$\hat{F}_{\kappa^*}(y \mid x) = \begin{cases} \dfrac{n+\tau}{N^{\kappa^*}_c + 1} & \text{si } y \in (C^{\kappa^*}_{(n)}, C^{\kappa^*}_{(n+1)}),\ n \in \{0, \ldots, N^{\kappa^*}_c\} \\ \dfrac{n' - 1 + (n'' - n' + 2)\tau}{N^{\kappa^*}_c + 1} & \text{si } y = C^{\kappa^*}_{(n)},\ n \in \{1, \ldots, N^{\kappa^*}_c\} \end{cases} \tag{3-37}$$

donde $n' = \min\{m : C^{\kappa^*}_{(m)} = C^{\kappa^*}_{(n)}\}$ y $n'' = \max\{m : C^{\kappa^*}_{(m)} = C^{\kappa^*}_{(n)}\}$ manejan la presencia de scores duplicados. El término de suavizado $\tau$ es crucial para garantizar que la CDF sea estrictamente creciente, evitando discontinuidades en presencia de empates.

**Garantías de Cobertura Local** La ventaja teórica del MCPS radica en su capacidad para lograr **cobertura condicional válida** dentro de cada estrato, no solo marginalmente. Boström, Johansson, and Löfström (2021) demuestran que, bajo intercambiabilidad dentro de cada bin, para cualquier nivel de significancia $\alpha$:

$$\mathbb{P}\left(Y \in \left[\hat{F}^{-1}_{\kappa}(\alpha/2 \mid x), \hat{F}^{-1}_{\kappa}(1-\alpha/2 \mid x)\right] \,\middle|\, x \in \text{Bin}_{\kappa}\right) \geq 1 - \alpha \tag{3-38}$$

Esto implica que las bandas de predicción se **ajustan automáticamente** a la heterogeneidad de la incertidumbre: regiones donde el modelo es más confiable producen intervalos estrechos, mientras que regiones con alta variabilidad residual generan intervalos más amplios. Formalmente, si denotamos $W_{\kappa}(x) = \hat{F}^{-1}_{\kappa}(1-\alpha/2 \mid x) - \hat{F}^{-1}_{\kappa}(\alpha/2 \mid x)$ como el ancho del intervalo de predicción, se cumple que:

$$W_{\kappa_1}(x_1) \neq W_{\kappa_2}(x_2) \quad \text{si} \quad \text{Var}(\epsilon \mid x \in \text{Bin}_{\kappa_1}) \neq \text{Var}(\epsilon \mid x \in \text{Bin}_{\kappa_2}) \tag{3-39}$$

Esta propiedad contrasta con el SCPS, donde $W(x) \approx$ constante para todo $x \in \mathcal{X}$, lo que puede resultar en sobre-cobertura (intervalos innecesariamente anchos) en regiones de baja incertidumbre o sub-cobertura (intervalos peligrosamente estrechos) en regiones de alta incertidumbre.

### De la Teoría a la Práctica

La implementación del MCPS para pronóstico de series temporales autorregresivas, desarrollada en esta investigación, constituye una **contribución metodológica novedosa** al traducir el framework teórico—originalmente concebido para datos multivariados independientes en problemas de logística (Ye, Hijazi, and Van Hentenryck 2025)—hacia el contexto de dependencias temporales. Esta extensión representa uno de los aportes más significativos de la presente tesis, dado que la aplicación del particionamiento Mondrian a series temporales no había sido previamente formalizada en la literatura conformal.

**Adaptación 1: Construcción Dinámica de Features Autorregresivos** **Teoría Original:** El trabajo seminal de Ye, Hijazi, and Van Hentenryck (2025) en logística de e-commerce asume que cada instancia $x_i$ es un vector de características pre-existentes (ubicación de almacén, transportista, hora del día, peso del paquete, etc.) observable al momento de la predicción. Estas features son estáticas en el sentido de que no dependen de predicciones previas.

**Adaptación para Series Temporales:** En esta tesis, los "objetos" $x_i$ no existen de forma independiente, sino que se construyen dinámicamente como ventanas deslizantes de $p$ rezagos:

$$x_t = [y_{t-p}, y_{t-p+1}, \ldots, y_{t-1}] \in \mathbb{R}^p \tag{3-40}$$

Esta transformación convierte la serie univariada en una matriz de diseño autoregresiva. Matemáticamente, si denotamos la serie temporal original como $\{y_t\}_{t=1}^{T}$, la matriz de diseño resultante es:

$$\mathbf{X} = \begin{bmatrix} y_1 & y_2 & \cdots & y_p \\ y_2 & y_3 & \cdots & y_{p+1} \\ \vdots & \vdots & \ddots & \vdots \\ y_{T-p} & y_{T-p+1} & \cdots & y_{T-1} \end{bmatrix} \in \mathbb{R}^{(T-p)\times p} \tag{3-41}$$

con vector objetivo correspondiente $\mathbf{y} = [y_{p+1}, y_{p+2}, \ldots, y_T]^T$.

**Implicación Crítica:** Mientras que en el contexto de e-commerce las features son intercambiables entre órdenes (una orden del lunes tiene el mismo tipo de información que una

orden del martes), en series temporales la matriz $\mathbf{X}$ exhibe dependencias inherentes entre filas. Esto plantea una pregunta teórica fundamental: ¿se preserva la validez del MCPS bajo autocorrelación?

La respuesta, aunque no demostrada formalmente en Boström, Johansson, and Löfström (2021), es afirmativa bajo condiciones de *mixing débil* (Yu 1994). Para procesos estacionarios $\alpha$-mixing con coeficiente de mixing $\alpha(k) \to 0$ cuando $k \to \infty$, las observaciones suficientemente separadas en el tiempo son "casi independientes", permitiendo que el conjunto de calibración $\mathcal{D}_c$ satisfaga aproximadamente el requisito de intercambiabilidad. Esta tesis proporciona evidencia empírica de que las garantías de cobertura se mantienen en la práctica para series estacionarias comunes.

**Adaptación 2: Estrategia de Binning Adaptativo** **Teoría Original:** Boström, Johansson, and Löfström (2021) utilizan cuantiles equiespaciados calculados sobre las predicciones de calibración $\{h(x_j)\}_{j=1}^{N_c}$ con $B$ bins fijos. Implícitamente, esto asume que las predicciones tienen soporte continuo con densidad suficiente en cada bin.

**Desafío Práctico:** En series temporales con baja variabilidad (procesos estacionarios alrededor de una media constante), las predicciones del modelo autorregresivo pueden concentrarse en un rango estrecho. Por ejemplo, si $h(x_j) \in [48, 52]$ para casi todos los $j$, intentar dividir este rango en $B = 10$ bins resultaría en múltiples bins con fronteras idénticas.

**Solución Implementada:** Se emplea una estrategia de *binning tolerante a duplicados*, donde bins con fronteras colapsadas se fusionan automáticamente. Matemáticamente, en lugar de imponer $B$ bins fijos, se permite que el número efectivo de bins sea:

$$B_{\text{efectivo}} = \left|\left\{q_{\frac{\kappa}{B}} : \kappa = 1, \ldots, B-1\right\}\right| \leq B \tag{3-42}$$

donde $|\cdot|$ denota cardinalidad del conjunto de cuantiles únicos. Esta adaptación es matemáticamente válida bajo el marco de Mondrian, ya que Vovk, Gammerman, and Shafer (2005) no requieren que todos los bins tengan el mismo tamaño, solo que la partición sea determinada a priori (antes de observar los datos de prueba).

**Mecanismo de Fallback:** Si el procedimiento de binning falla completamente (por ejemplo, en series perfectamente constantes donde todas las predicciones son idénticas), el sistema degrada automáticamente a SCPS global, utilizando todo $\mathcal{D}_c$ sin particionamiento.

Este fallback garantiza robustez operativa sin violar los principios teóricos conformales.

**Adaptación 3: Representación Discreta de la Distribución Predictiva Desviación Fundamental de la Teoría:** El trabajo de Ye, Hijazi, and Van Hentenryck (2025) utiliza la librería `Crepes` (Boström 2022), que implementa la ecuación (5) para generar una función de distribución acumulada (CDF) continua por partes. Esta representación es conveniente para cálculos de cuantiles arbitrarios pero requiere almacenar la función completa.

**Enfoque Adoptado en esta Tesis:** Se retorna directamente el conjunto de *calibration scores* $\{C_j^{\kappa^*}\}_{j=1}^{N_c^{\kappa^*}}$ sin suavizado ni transformación adicional. Esta decisión representa una de las contribuciones metodológicas más importantes de la investigación y merece análisis detallado.

**Justificación Teórica Profunda:** Según Vovk, Gammerman, and Shafer (2022), los scores conformales $\{C_j^{\kappa^*}\}$ constituyen una muestra válida de la *distribución predictiva empírica exacta* bajo el supuesto de intercambiabilidad. Formalmente, si denotamos $\mathcal{C}^{\kappa^*} = \{C_1^{\kappa^*}, \ldots, C_{N_c^{\kappa^*}}^{\kappa^*}\}$ como el conjunto de scores locales, entonces para cualquier función medible $g$:

$$\mathbb{E}[g(Y) \mid x \in \text{Bin}_{\kappa^*}] \approx \frac{1}{N_c^{\kappa^*}} \sum_{j=1}^{N_c^{\kappa^*}} g(C_j^{\kappa^*}) \tag{3-43}$$

con error de aproximación que converge a cero cuando $N_c^{\kappa^*} \to \infty$. Esto significa que cualquier estadístico de interés (media, varianza, cuantiles, etc.) puede calcularse directamente sobre $\mathcal{C}^{\kappa^*}$ sin necesidad de reconstruir la CDF completa.

**Ventaja Computacional:** Esta representación discreta es especialmente eficiente para el cálculo del *Continuous Ranked Probability Score* (CRPS), que requiere integrar sobre toda la distribución predictiva:

$$\text{CRPS}(\hat{F}, y) = \int_{-\infty}^{\infty} \left[\hat{F}(u) - \mathbb{1}\{u \geq y\}\right]^2 du \tag{3-44}$$

Para una distribución empírica discreta representada por $\mathcal{C}^{\kappa^*}$, esta integral se reduce a:

$$\text{CRPS}(\mathcal{C}^{\kappa^*}, y) = \frac{1}{N_c^{\kappa^*}} \sum_{j=1}^{N_c^{\kappa^*}} |C_j^{\kappa^*} - y| - \frac{1}{2(N_c^{\kappa^*})^2} \sum_{j=1}^{N_c^{\kappa^*}} \sum_{k=1}^{N_c^{\kappa^*}} |C_j^{\kappa^*} - C_k^{\kappa^*}| \tag{3-45}$$

lo cual es computacionalmente más eficiente que evaluar la integral sobre la CDF continua suavizada.

**Comparación con LSPM:** Esta filosofía de representación alinea el MCPS con la estrategia implementada para LSPM (ver sección anterior), donde los valores críticos $C_i$ se retornan sin procesamiento adicional. Ambos modelos convergen así hacia una representación purista de la teoría conformal, evitando artificialidades que no están respaldadas teóricamente y que podrían degradar las propiedades de cobertura finita.

**Adaptación 4: Asignación de Bins para Puntos de Prueba** Para una nueva observación $x_{\text{test}}$ con predicción puntual $h(x_{\text{test}})$, la asignación al bin correspondiente se realiza mediante búsqueda binaria sobre los bordes de bins pre-calculados $\{q_{\kappa/B}\}_{\kappa=0}^{B}$:

$$\kappa^* = \min \left\{ \kappa \in \{1, \ldots, B\} : h(x_{\text{test}}) < q_{\kappa/B} \right\} \tag{3-46}$$

Esta operación tiene complejidad temporal $O(\log B)$, significativamente más eficiente que la búsqueda lineal $O(B)$.

**Mecanismo de Fallback Jerárquico:** Si el bin localizado $\kappa^*$ contiene menos de un umbral mínimo de observaciones de calibración (típicamente 5), el sistema degrada a SCPS global:

$$\mathcal{D}_c^{\text{efectivo}} = \begin{cases} \mathcal{D}_c^{\kappa^*} & \text{si } N_c^{\kappa^*} \geq 5 \\ \mathcal{D}_c & \text{si } N_c^{\kappa^*} < 5 \end{cases} \tag{3-47}$$

Esta heurística, aunque **no está especificada en la teoría original** de Boström, Johansson, and Löfström (2021), resulta esencial en la práctica para prevenir intervalos de predicción erráticamente anchos cuando el tamaño de muestra local es insuficiente. Teóricamente, puede interpretarse como un estimador shrinkage que balancea sesgo (usar todos los datos) contra varianza (usar solo datos locales).

### Diferencias Notables Respecto a la Teoría Original

La Tabla **3-4** resume las principales diferencias entre el marco teórico propuesto por Ye, Hijazi, and Van Hentenryck (2025) y la implementación desarrollada en esta tesis.

| **Aspecto** | **Teoría (Ye et al., 2025)** | **Implementación (Esta Tesis)** |
|---|---|---|
| Dominio de aplicación | Logística de e-commerce (órdenes independientes) | Series temporales (dependencias autorregresivas) |
| Construcción de features | Pre-existentes ($x_i$) observables | Dinámicas (ventanas deslizantes de rezagos) |
| Librería conformal | `Crepes` (Boström 2022) | Implementación directa de ecuaciones teóricas |
| Representación de CPD | CDF continua por partes con suavizado $\tau$ | Distribución empírica discreta exacta |
| Binning robusto | Cuantiles fijos, no se menciona manejo de duplicados | Fusión automática de bins con fronteras colapsadas |
| Fallback a SCPS | No mencionado en el artículo | Automático si $N_c^\kappa < 5$ |
| Validación de intercambiabilidad | Asumida para órdenes independientes | Verificada empíricamente bajo mixing débil |
| Horizonte de predicción | Batch (todas las predicciones simultáneas) | Secuencial (rolling forecast paso a paso) |

Table **3-4**: Comparación entre MCPS Teórico y MCPS para Series Temporales

**Innovación Metodológica: MCPS Autorregresivo** La contribución más significativa de esta implementación es la **extensión del framework Mondrian a series temporales autorregresivas**, lo cual no había sido previamente formalizado en la literatura. Mientras que Boström, Johansson, and Löfström (2021) demuestran la validez teórica del particionamiento para datos i.i.d., esta tesis proporciona evidencia tanto teórica como empírica de que las garantías de cobertura se preservan bajo dependencias temporales débiles.

Formalmente, para un proceso estacionario $\{y_t\}$ que satisface:

$$\sup_{A\in\mathcal{F}_1^k, B\in\mathcal{F}_{k+n}^\infty} |P(A\cap B) - P(A)P(B)| = \alpha(n) \to 0 \quad \text{cuando } n\to\infty \tag{3-48}$$

donde $\mathcal{F}_a^b$ denota la $\sigma$-álgebra generada por $\{y_t : t \in [a,b]\}$, se puede demostrar que el conjunto de calibración $\mathcal{D}_c$ satisface intercambiabilidad aproximada con error $O(\alpha(n))$. Esto implica que la pérdida de cobertura del MCPS autorregresivo es acotada superiormente por:

$$\left|\mathbb{P}(Y \in \hat{C}_\alpha(x)) - (1-\alpha)\right| \leq C\cdot\alpha(n) + o(N_c^{-1/2}) \tag{3-49}$$

donde $C$ es una constante que depende de las características del proceso y $\hat{C}_\alpha(x)$ denota el intervalo de predicción conformal.

Esta extensión abre la puerta a futuras investigaciones sobre predicción conformal para procesos no estacionarios, potencialmente integrando esquemas de ponderación temporal (similar a LSPMW presentado anteriormente) con particionamiento Mondrian adaptativo. Un desarrollo promisorio sería el *Weighted Mondrian CPS*, donde cada bin $\kappa$ tendría pesos decayentes $w_j^\kappa$ asociados a sus scores de calibración, permitiendo adaptación simultánea tanto a heterogeneidad espacial (via binning) como temporal (via ponderación).

### Parámetros e Hiperparámetros en la Implementación

**Hiperparámetros Primarios n_lags ($p$):**

- *Tipo:* Entero positivo
- *Rango típico:* $\{5, 10, 15, 20\}$
- *Significado:* Define el orden del modelo autorregresivo subyacente. Controla cuánta memoria temporal incorpora el predictor.
- *Selección:* Valor por defecto $p = 10$, balanceando la captura de patrones cíclicos con el riesgo de sobreajuste. Para series con alta autocorrelación (procesos AR de orden alto), valores mayores como $p = 15$ o $p = 20$ pueden ser apropiados.

**n_bins ($B$):**

- *Tipo:* Entero positivo
- *Rango teórico recomendado:* Boström, Johansson, and Löfström (2021) sugieren $B \in \{5, 10, 15\}$ para balancear adaptabilidad local contra varianza por tamaño de muestra.
- *Valor por defecto:* $B = 10$, coincidiendo con la configuración de Ye, Hijazi, and Van Hentenryck (2025).
- *Impacto en cobertura:* Valores pequeños ($B \leq 3$) degradan el MCPS hacia SCPS, perdiendo la adaptabilidad local. Valores excesivos ($B \geq 20$) fragmentan demasiado el conjunto de calibración, violando el requisito de tamaño mínimo de muestra en bins extremos.
- *Relación matemática:* El tamaño esperado de cada bin es $\mathbb{E}[N_c^\kappa] = N_c/B$, asumiendo uniformidad en las predicciones, aunque en la práctica los bins raramente son equipoblados debido a concentración de predicciones en regiones específicas.

**test_size:**

- *Tipo:* Proporción en $(0, 1)$
- *Valor por defecto:* 0.25
- *Función:* Define el split entre conjunto de entrenamiento propio y conjunto de calibración conforme. Formalmente:

$$N_c = \lfloor \text{test\_size} \times N_{\text{total}} \rfloor \tag{3-50}$$

- *Trade-off:* Valores altos ($> 0.3$) proveen más residuos de calibración para estimar la CPD, mejorando la precisión de los cuantiles, pero reducen los datos disponibles para entrenar el modelo base $h(x)$, potencialmente degradando la calidad de las predicciones puntuales. Valores bajos ($< 0.15$) inducen distribuciones predictivas ruidosas por escasez de scores.

**Parámetros del Modelo Base** El modelo autorregresivo subyacente $h : \mathbb{R}^p \to \mathbb{R}$ se implementa mediante XGBoost (T. Chen and Guestrin 2016) con configuración conservadora: 50 árboles de profundidad 3 y tasa de aprendizaje 0.1. Esta parametrización se justifica por:

- **Profundidad limitada (max_depth = 3):** Previene sobreajuste al limitar la complejidad individual de cada árbol, crítico cuando $p$ es grande relativo a $N_{\text{train}}$. Teóricamente, árboles poco profundos actúan como modelos aditivos generalizados (GAMs) que capturan efectos principales sin interacciones de orden alto.
- **Número moderado de árboles ($n = 50$):** Ye, Hijazi, and Van Hentenryck (2025) reportan rendimiento similar con $n \in \{50, 100\}$ en su dominio de logística. Para series temporales, valores excesivos pueden capturar ruido en lugar de patrones genuinos.
- **Tasa de aprendizaje estándar ($\eta = 0.1$):** Permite convergencia estable del algoritmo de boosting sin requerir early stopping.

**Diferencia con la Teoría:** El artículo de Ye, Hijazi, and Van Hentenryck (2025) no especifica la arquitectura del modelo base, simplemente requiere que $h : \mathcal{X} \to \mathbb{R}$ sea cualquier regressor. La elección de XGBoost en esta tesis se justifica por su robustez a multicolinealidad inherente en features autorregresivos y su eficiencia computacional.

**Interpretación Práctica de los Hiperparámetros** La selección conjunta de $(p, B, \text{test\_size})$ define un balance fundamental en el MCPS:

$$\text{Calidad de } h(x) \propto (1 - \text{test\_size}) \cdot f(p) \tag{3-51}$$

$$\text{Precisión de CPD} \propto \text{test\_size} \cdot \frac{N_c}{B} \tag{3-52}$$

donde $f(p)$ es una función no monótona que inicialmente crece con $p$ (mayor capacidad de captura de patrones) pero eventualmente decrece por sobreajuste. El hiperparámetro $B$ actúa como regulador de la localidad: valores grandes producen adaptación fina pero requieren $N_c$ grande para mantener $N_c/B$ suficiente en cada bin.

Para series temporales típicas con $T \approx 1000$ observaciones, la configuración estándar $(p = 10, B = 10, \text{test\_size} = 0.25)$ resulta en aproximadamente 25 scores de calibración por bin, que es suficiente para estimación robusta de cuantiles según Rob J. Hyndman and Fan (1996).

### 3.4.6 Adaptive Volatility Mondrian Conformal Predictive System (AV-MCPS)

#### Propuesta Teórica

El *Adaptive Volatility Mondrian Conformal Predictive System* (AV-MCPS) constituye una extensión metodológica del MCPS estándar desarrollada específicamente para esta investigación, y representa una de las contribuciones originales más significativas de la presente tesis. Mientras que el MCPS convencional de Boström, Johansson, and Löfström (2021) particiona el espacio de calibración únicamente en función de las predicciones puntuales $h(x)$, el AV-MCPS introduce una **estratificación bidimensional** que incorpora explícitamente la volatilidad local como segunda dimensión de heterogeneidad.

**Motivación: Límites del Particionamiento Unidimensional** El particionamiento Mondrian estándar asume implícitamente que la variabilidad del error de predicción es homogénea dentro de cada bin de predicción. Formalmente, si $\mathcal{D}_c^\kappa$ denota el subconjunto de calibración con predicciones en el rango $[q_{\kappa/B}, q_{(\kappa+1)/B})$, el MCPS supone:

$$\text{Var}(\epsilon_i \mid h(x_i) \in \mathcal{D}_c^\kappa) \approx \text{constante} \quad \forall i \in \mathcal{D}_c^\kappa \tag{3-53}$$

Sin embargo, esta suposición es frecuentemente violada en series temporales que exhiben **heterocedasticidad condicional**, donde la volatilidad de los errores varía sistemáticamente en el tiempo. Procesos como GARCH, modelos de volatilidad estocástica, o simplemente series con períodos de alta y baja turbulencia, presentan estructuras de error donde:

$$\text{Var}(\epsilon_t) = \sigma_t^2 \neq \text{constante} \tag{3-54}$$

En estos contextos, dos observaciones con predicciones puntuales similares $h(x_i) \approx h(x_j)$ pueden experimentar errores de magnitudes radicalmente diferentes si provienen de regímenes de volatilidad distintos. El MCPS estándar, al ignorar esta dimensión, podría producir distribuciones predictivas mal calibradas: demasiado estrechas en períodos de alta volatilidad (sub-cobertura) y excesivamente anchas en períodos de baja volatilidad (sobre-cobertura).

**Fundamento Teórico: Particionamiento Bidimensional** El AV-MCPS propone una partición conjunta del espacio de calibración basada en dos características simultáneas:

1. **Predicción puntual** $h(x_i)$: Captura el nivel esperado de la variable objetivo, similar al MCPS estándar.
2. **Volatilidad local** $\sigma_i$: Mide la variabilidad reciente del proceso en una ventana temporal anterior a la observación $i$.

Formalmente, sea $V : \mathbb{N} \to \mathbb{R}^+$ una función que mapea cada índice temporal $i$ a su volatilidad estimada $\sigma_i$, calculada mediante una ventana rodante de longitud $w$:

$$\sigma_i = \sqrt{\frac{1}{w-1} \sum_{k=i-w}^{i-1} (y_k - \bar{y}_{i,w})^2} \tag{3-55}$$

donde $\bar{y}_{i,w}$ es la media muestral en la ventana $[i-w, i-1]$. El conjunto de calibración se particiona entonces en una grilla bidimensional:

$$\mathcal{D}_c = \bigcup_{\kappa=1}^{B_{\text{pred}}} \bigcup_{\lambda=1}^{B_{\text{vol}}} \mathcal{D}_c^{(\kappa,\lambda)} \tag{3-56}$$

donde cada celda $\mathcal{D}_c^{(\kappa,\lambda)}$ agrupa observaciones que satisfacen simultáneamente:

$$\mathcal{D}_c^{(\kappa,\lambda)} = \left\{ (x_i, y_i) \in \mathcal{D}_c : \begin{array}{l} q_{\text{pred}}^{\kappa-1} \leq h(x_i) < q_{\text{pred}}^{\kappa} \\ \text{y } q_{\text{vol}}^{\lambda-1} \leq \sigma_i < q_{\text{vol}}^{\lambda} \end{array} \right\} \tag{3-57}$$

Aquí, $q_{\text{pred}}^{\kappa}$ son los cuantiles de las predicciones de calibración y $q_{\text{vol}}^{\lambda}$ son los cuantiles de las volatilidades de calibración. Esta estratificación genera un total de $B_{\text{pred}} \times B_{\text{vol}}$ celdas, cada una representando un régimen específico de (nivel, volatilidad).

**Intuición del Particionamiento Bidimensional** La lógica del AV-MCPS puede ilustrarse mediante un ejemplo concreto. Consideremos dos instancias de calibración:

- Observación $A$: $h(x_A) = 50$, $\sigma_A = 2$ (predicción media, baja volatilidad)
- Observación $B$: $h(x_B) = 51$, $\sigma_B = 8$ (predicción media, alta volatilidad)

El MCPS estándar, al depender únicamente de $h(x)$, asignaría ambas observaciones al mismo bin de predicción, asumiendo que sus distribuciones de error son intercambiables. Sin embargo, el AV-MCPS reconoce que la observación $B$ proviene de un régimen de mayor incertidumbre intrínseca, donde los errores de predicción tienden a ser más grandes en magnitud absoluta. Por lo tanto, las asigna a celdas distintas: $A \in \mathcal{D}_c^{(\kappa,\lambda_{\text{low}})}$ y $B \in \mathcal{D}_c^{(\kappa,\lambda_{\text{high}})}$.

Cuando se genera la distribución predictiva para un nuevo punto de prueba $x_{\text{test}}$ con predicción $h(x_{\text{test}}) = 50.5$ y volatilidad actual $\sigma_{\text{test}} = 7.5$, el AV-MCPS utiliza calibration scores exclusivamente de la celda $\mathcal{D}_c^{(\kappa,\lambda_{\text{high}})}$, produciendo una distribución predictiva naturalmente más ancha que reflejará la mayor incertidumbre del régimen de alta volatilidad.

**Cálculo Localizado de Scores Conformales** El procedimiento de inferencia del AV-MCPS sigue estos pasos:

1. Para un punto de prueba $x_{\text{test}}$, calcular su predicción puntual $h(x_{\text{test}})$ y su volatilidad local $\sigma_{\text{test}}$ usando el modelo base y la ventana temporal disponible.
2. Determinar la celda correspondiente $(\kappa^*, \lambda^*)$ mediante:

$$\kappa^* = \arg\min_{\kappa} \left\{ \kappa : h(x_{\text{test}}) < q_{\text{pred}}^{\kappa} \right\} \tag{3-58}$$

$$\lambda^* = \arg\min_{\lambda} \left\{ \lambda : \sigma_{\text{test}} < q_{\text{vol}}^{\lambda} \right\} \tag{3-59}$$

3. Extraer el subconjunto local de calibración:

$$\mathcal{D}_c^{(\kappa^*,\lambda^*)} = \left\{ (x_i, y_i) : i \in \text{Bin}_{\kappa^*}^{\text{pred}} \cap \text{Bin}_{\lambda^*}^{\text{vol}} \right\} \tag{3-60}$$

4. Calcular calibration scores localizados:

$$C_i^{(\kappa^*,\lambda^*)} = h(x_{\text{test}}) + (y_i - h(x_i)), \quad \forall (x_i, y_i) \in \mathcal{D}_c^{(\kappa^*,\lambda^*)} \tag{3-61}$$

La distribución predictiva se construye entonces a partir de estos scores localizados, de manera idéntica al MCPS estándar.

**Garantías Teóricas de Cobertura** Bajo el supuesto de intercambiabilidad condicional dentro de cada celda, es decir:

$$(x_i, y_i) \mid h(x_i) \in \text{Bin}_\kappa, \sigma_i \in \text{Bin}_\lambda \stackrel{d}{=} (x_j, y_j) \mid h(x_j) \in \text{Bin}_\kappa, \sigma_j \in \text{Bin}_\lambda \tag{3-62}$$

para todo $i, j$, las garantías de cobertura conformal se mantienen localmente:

$$\mathbb{P}\left(Y \in \hat{C}_\alpha(x) \,\middle|\, x \in \text{Bin}_\kappa^{\text{pred}}, \sigma(x) \in \text{Bin}_\lambda^{\text{vol}}\right) \geq 1 - \alpha \tag{3-63}$$

Esta garantía es más fuerte que la del MCPS estándar, ya que condiciona sobre una partición más fina del espacio. Intuitivamente, al controlar simultáneamente por nivel y volatilidad, el AV-MCPS logra una **adaptabilidad condicional mejorada**, reduciendo la heterogeneidad residual dentro de cada celda.

**Trade-off: Resolución vs. Tamaño de Muestra** El particionamiento bidimensional introduce un trade-off fundamental. Sea $N_c$ el tamaño total del conjunto de calibración. El número esperado de observaciones por celda es:

$$\mathbb{E}\left[N_c^{(\kappa,\lambda)}\right] = \frac{N_c}{B_{\text{pred}} \times B_{\text{vol}}} \tag{3-64}$$

asumiendo distribución uniforme (que raramente se cumple en la práctica). Aumentar la resolución del particionamiento ($B_{\text{pred}}, B_{\text{vol}} \uparrow$) mejora la localización pero reduce el tamaño de muestra en cada celda, incrementando la varianza de los cuantiles estimados. Este trade-off es más severo que en MCPS estándar debido al producto de dimensiones.

Para mitigar este problema, el AV-MCPS implementa una **estrategia de fallback jerárquico** que degrada progresivamente la localización cuando el tamaño de muestra es insuficiente:

$$\mathcal{D}_c^{\text{efectivo}} = \begin{cases} \mathcal{D}_c^{(\kappa^*,\lambda^*)} & \text{si } N_c^{(\kappa^*,\lambda^*)} \geq \tau_{\min} \\ \mathcal{D}_c^{(\kappa^*,\cdot)} & \text{si } N_c^{(\kappa^*,\lambda^*)} < \tau_{\min} \text{ y } N_c^{(\kappa^*,\cdot)} \geq \tau_{\min} \\ \mathcal{D}_c & \text{en otro caso} \end{cases} \tag{3-65}$$

donde $\mathcal{D}_c^{(\kappa^*,\cdot)} = \bigcup_{\lambda=1}^{B_{\text{vol}}} \mathcal{D}_c^{(\kappa^*,\lambda)}$ representa el bin unidimensional basado solo en predicción,

y $\tau_{\min}$ es un umbral de tamaño mínimo (típicamente 5 observaciones). Esta estrategia garantiza robustez operativa sin sacrificar las garantías teóricas conformales.

### De la Teoría a la Práctica

La implementación del AV-MCPS para series temporales autorregresivas, desarrollada en esta tesis, traduce el marco teórico bidimensional en un sistema predictivo operativo mediante las siguientes decisiones de diseño.

**Adaptación 1: Estimación Rolling de Volatilidad** **Desafío Práctico:** En series temporales reales, la volatilidad $\sigma_t$ no es directamente observable y debe estimarse a partir de datos históricos. Diferentes métodos de estimación (desviación estándar simple, GARCH, volatilidad realizada, etc.) pueden producir señales de volatilidad con características distintas.

**Solución Implementada:** Se emplea la *desviación estándar rolling* con ventana de longitud $w$, calculada mediante:

$$\hat{\sigma}_t = \sqrt{\frac{1}{w-1}\sum_{k=t-w}^{t-1}(y_k - \bar{y}_t)^2} \tag{3-66}$$

donde $\bar{y}_t = \frac{1}{w}\sum_{k=t-w}^{t-1} y_k$ es la media rolling. Esta elección se justifica por su:

- **Simplicidad computacional:** No requiere ajuste iterativo de parámetros como GARCH.
- **Robustez:** La ventana fija previene la amplificación de shocks transitorios.
- **Interpretabilidad:** Representa directamente la dispersión histórica reciente.

**Manejo de Valores Faltantes:** Para las primeras $w-1$ observaciones donde la ventana completa no está disponible, se aplica *backfilling* con la desviación estándar de toda la serie inicial disponible:

$$\hat{\sigma}_t = \sqrt{\frac{1}{t-2}\sum_{k=1}^{t-1}(y_k - \bar{y}_{1:t-1})^2} \quad \text{para } t < w \tag{3-67}$$

Esto asegura que todas las observaciones de calibración tengan una volatilidad asociada, evitando pérdida de información.

**Adaptación 2: Construcción de Bins mediante Cuantiles Robustos** **Teoría Original:** El particionamiento Mondrian estándar utiliza cuantiles equiespaciados calculados sobre las predicciones de calibración. Esta estrategia asume implícitamente que las predicciones tienen distribución continua con densidad suficiente en todo el soporte.

**Desafío en Volatilidad:** Las series de volatilidad estimada $\{\hat{\sigma}_i\}$ frecuentemente exhiben distribuciones asimétricas con colas pesadas (outliers durante períodos de crisis) y concentración de masa en valores bajos. Intentar dividir esta distribución en $B_{\text{vol}}$ bins equiespaciados puede resultar en bins con fronteras colapsadas.

**Solución Implementada:** Se emplea la función `pd.qcut` con opción `duplicates='drop'`, que fusiona automáticamente bins con fronteras idénticas. El número efectivo de bins de volatilidad es:

$$B_{\text{vol}}^{\text{efectivo}} = \left|\left\{q_{\text{vol}}^{\lambda} : \lambda = 1, \ldots, B_{\text{vol}} - 1\right\}\right| \leq B_{\text{vol}} \tag{3-68}$$

donde $|\cdot|$ denota cardinalidad del conjunto de cuantiles únicos. Esta adaptación es matemáticamente válida bajo el marco conformal, ya que no requiere que todos los bins tengan el mismo tamaño, solo que la partición sea determinada usando únicamente datos de calibración.

**Mecanismo de Fallback:** Si el procedimiento de binning falla completamente en alguna dimensión (por ejemplo, todas las predicciones idénticas o volatilidad constante), el sistema degrada automáticamente a MCPS unidimensional usando solo la dimensión válida, o a SCPS global si ambas dimensiones fallan.

**Adaptación 3: Representación Directa de Calibration Scores** Consistente con la filosofía adoptada para LSPM y MCPS, el AV-MCPS retorna directamente el conjunto de calibration scores sin transformación adicional:

$$\mathcal{C}^{(\kappa^*,\lambda^*)} = \left\{C_i^{(\kappa^*,\lambda^*)}\right\}_{i=1}^{N_c^{(\kappa^*,\lambda^*)}} \tag{3-69}$$

Esta representación empírica discreta constituye una muestra válida de la distribución

predictiva conformal bajo intercambiabilidad. Cualquier estadístico de interés (media, varianza, cuantiles, CRPS) puede calcularse directamente sobre $\mathcal{C}^{(\kappa^*,\lambda^*)}$ sin necesidad de reconstruir una CDF continua, manteniendo pureza teórica y eficiencia computacional.

**Adaptación 4: Protocolo de Congelamiento Bidimensional** Para garantizar rigor estadístico en la evaluación rolling, el AV-MCPS implementa un protocolo de congelamiento que fija simultáneamente:

1. **Modelo base:** Los parámetros del regressor XGBoost se entrenan una sola vez sobre el conjunto de entrenamiento y se congelan.
2. **Bins de predicción:** Los bordes $\{q^{\kappa}_{\text{pred}}\}_{\kappa=0}^{B_{\text{pred}}}$ se calculan sobre las predicciones de calibración y se congelan.
3. **Bins de volatilidad:** Los bordes $\{q^{\lambda}_{\text{vol}}\}_{\lambda=0}^{B_{\text{vol}}}$ se calculan sobre las volatilidades de calibración y se congelan.
4. **Artefactos de calibración:** Las predicciones $\{h(x_i)\}$, valores observados $\{y_i\}$, y volatilidades $\{\sigma_i\}$ del conjunto de calibración se almacenan.

Durante la fase de evaluación rolling, para cada nuevo punto de prueba $t$:

- Se extraen los últimos $p$ valores observados para construir $x_t = [y_{t-p}, \ldots, y_{t-1}]$
- Se calcula $h(x_t)$ usando el modelo congelado
- Se estima $\sigma_t$ usando la ventana rolling actual
- Se asigna $(h(x_t), \sigma_t)$ a la celda $(\kappa^*, \lambda^*)$ usando los bordes congelados
- Se generan scores conformales usando los artefactos de calibración congelados

Este protocolo es crítico para prevenir data leakage y garantizar que la evaluación del desempeño predictivo sea válida.

### Diferencias Respecto al MCPS Estándar

La Tabla **3-5** resume las principales diferencias metodológicas entre MCPS y AV-MCPS.

| **Aspecto** | **MCPS** | **AV-MCPS** |
|---|---|---|
| Dimensiones de particionamiento | Unidimensional (solo predicción) | Bidimensional (predicción + volatilidad) |
| Número de bins | $B$ | $B_{\text{pred}} \times B_{\text{vol}}$ |
| Tamaño esperado de celda | $N_c/B$ | $N_c/(B_{\text{pred}} \times B_{\text{vol}})$ |
| Captura de heterocedasticidad | Indirecta (via niveles de predicción) | Explícita (via volatilidad local) |
| Complejidad computacional | $O(\log B)$ por predicción | $O(\log B_{\text{pred}} + \log B_{\text{vol}})$ |
| Estrategia de fallback | Degradar a SCPS si $N_c^{\kappa} < \tau$ | Jerárquica: 2D $\rightarrow$ 1D $\rightarrow$ SCPS |
| Hiperparámetros adicionales | Ninguno | `volatility_window` $(w)$, $B_{\text{vol}}$ |
| Casos de uso óptimos | Heterogeneidad determinada por nivel | Series con volatilidad cambiante en el tiempo |

Table **3-5**: Comparación entre MCPS y AV-MCPS

**Innovación Metodológica: Estratificación Volatilidad-Predicción** La contribución fundamental del AV-MCPS es el reconocimiento de que la **volatilidad local es un predictor del error de predicción independiente del nivel de predicción**. Formalmente, si denotamos el error absoluto de predicción como $e_i = |y_i - h(x_i)|$, la hipótesis subyacente del AV-MCPS es:

$$\mathbb{E}[e_i \mid h(x_i), \sigma_i] \neq \mathbb{E}[e_i \mid h(x_i)] \tag{3-70}$$

Es decir, conocer la volatilidad local $\sigma_i$ provee información adicional sobre la magnitud esperada del error, más allá de la predicción puntual $h(x_i)$. Evidencia empírica de esta relación se encuentra ampliamente documentada en la literatura de volatilidad condicional (Engle 1982; Bollerslev 1986).

El AV-MCPS explota esta estructura mediante la estratificación bidimensional, logrando distribuciones predictivas que se adaptan simultáneamente tanto al *nivel* como al *régimen de incertidumbre* del proceso. Esta adaptabilidad dual representa una ventaja teórica significativa sobre el MCPS unidimensional, especialmente en series con volatilidad time-varying como procesos financieros, climáticos, o epidemiológicos.

### Parámetros e Hiperparámetros en la Implementación

**Hiperparámetros Primarios** **n_lags ($p$):**

- *Tipo:* Entero positivo
- *Valor por defecto:* $p = 15$, ligeramente mayor que MCPS estándar para capturar dependencias de más largo plazo
- *Significado:* Define el orden del modelo autorregresivo subyacente

**n_pred_bins ($B_{\text{pred}}$):**

- *Tipo:* Entero positivo
- *Valor por defecto:* $B_{\text{pred}} = 8$
- *Rango recomendado:* $[5, 10]$
- *Función:* Controla la resolución del particionamiento en la dimensión de predicción

**n_vol_bins ($B_{\mathbf{vol}}$):**

- *Tipo:* Entero positivo
- *Valor por defecto:* $B_{\text{vol}} = 4$
- *Rango recomendado:* $[3, 5]$
- *Función:* Controla la resolución del particionamiento en la dimensión de volatilidad
- *Justificación de valor menor:* La distribución de volatilidades tiende a ser más concentrada que las predicciones, requiriendo menos bins para capturar los regímenes principales (baja, media, alta volatilidad)

**volatility_window ($w$):**

- *Tipo:* Entero positivo
- *Valor por defecto:* $w = 20$
- *Rango típico:* $[10, 30]$
- *Función:* Define la longitud de la ventana rolling para estimar volatilidad local
- *Trade-off:* Ventanas pequeñas capturan cambios rápidos de volatilidad pero son más ruidosas; ventanas grandes suavizan estimaciones pero reaccionan lentamente a cambios de régimen

**test_size:**

- *Tipo:* Proporción en $(0, 1)$
- *Valor por defecto:* 0.25
- *Consideración especial:* Dado que el particionamiento bidimensional fragmenta más el conjunto de calibración, valores menores a 0.20 pueden resultar en celdas demasiado dispersas

**Parámetros del Modelo Base** El modelo autorregresivo subyacente se implementa idénticamente al MCPS estándar mediante XGBoost con 50 árboles de profundidad 3 y tasa de aprendizaje 0.1.

**Interpretación Práctica de la Configuración Bidimensional** La selección conjunta de $(B_{\text{pred}}, B_{\text{vol}}, N_c)$ determina el balance fundamental del AV-MCPS. El tamaño esperado de celda es:

$$\mathbb{E}[N_c^{(\kappa,\lambda)}] = \frac{N_c}{B_{\text{pred}} \times B_{\text{vol}}} \tag{3-71}$$

Para la configuración estándar ($B_{\text{pred}} = 8, B_{\text{vol}} = 4, N_c = 50$) típica en series con $T \approx 250$ observaciones, esto resulta en:

$$\mathbb{E}[N_c^{(\kappa,\lambda)}] = \frac{50}{8 \times 4} = 1.56 \text{ observaciones por celda} \tag{3-72}$$

Este valor es evidentemente insuficiente para estimación robusta, lo que justifica la necesidad crítica del mecanismo de fallback jerárquico. En la práctica, la distribución de observaciones entre celdas es altamente no uniforme: celdas centrales (predicción y volatilidad media) contienen muchas más observaciones que celdas extremas.

**Recomendaciones de Configuración** Para series temporales con $T \geq 500$ observaciones:

- Configuración balanceada: $(B_{\text{pred}} = 8, B_{\text{vol}} = 4, w = 20)$
- Para series con volatilidad muy cambiante: aumentar $B_{\text{vol}} = 5$ y reducir $w = 15$
- Para series estables: reducir $B_{\text{vol}} = 3$ y aumentar $w = 30$

Para series con $T < 300$ observaciones:

- Configuración conservadora: $(B_{\text{pred}} = 6, B_{\text{vol}} = 3, w = 15)$
- Considerar degradar a MCPS estándar si $T < 200$

### 3.4.7 DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks

#### Propuesta Teórica

DeepAR, introducido por Salinas et al. (2020), representa un cambio de paradigma en el pronóstico probabilístico de series temporales al trasladar el enfoque desde el modelado

individual de cada serie hacia el **aprendizaje de un modelo global** a partir de múltiples series relacionadas. A diferencia de los métodos clásicos basados en la metodología Box-Jenkins (Box and Jenkins 1968) o suavizamiento exponencial (Rob J. Hyndman, Koehler, et al. 2008), donde los parámetros se estiman independientemente para cada serie, DeepAR aprovecha la información compartida entre series mediante una arquitectura de red neuronal recurrente autorregresiva.

**Fundamento Arquitectónico: Redes LSTM Autorregresivas** El modelo se basa en una red neuronal recurrente con celdas LSTM (Long Short-Term Memory) que procesa secuencias temporales de forma autorregresiva. Formalmente, para una serie temporal $i$ con valores $z_{i,t}$ en el tiempo $t$, DeepAR modela la distribución condicional del futuro dado el pasado:

$$P(z_{i,t_0:T} \mid z_{i,1:t_0-1}, x_{i,1:T}) \tag{3-73}$$

donde $t_0$ denota el punto de inicio del horizonte de predicción, $z_{i,1:t_0-1}$ representa el rango de condicionamiento (valores observados), $z_{i,t_0:T}$ representa el rango de predicción (valores futuros), y $x_{i,1:T}$ son covariables conocidas para todo el período.

La arquitectura factoriza esta distribución mediante el producto de verosimilitudes condicionales:

$$Q_\Theta(z_{i,t_0:T} \mid z_{i,1:t_0-1}, x_{i,1:T}) = \prod_{t=t_0}^{T} Q_\Theta(z_{i,t} \mid z_{i,1:t-1}, x_{i,1:T}) \tag{3-74}$$

donde cada factor está parametrizado por la salida de la red recurrente:

$$Q_\Theta(z_{i,t} \mid z_{i,1:t-1}, x_{i,1:T}) = \ell(z_{i,t} \mid \theta(h_{i,t}, \Theta)) \tag{3-75}$$

El estado oculto $h_{i,t}$ se actualiza recursivamente mediante:

$$h_{i,t} = h(h_{i,t-1}, z_{i,t-1}, x_{i,t}, \Theta) \tag{3-76}$$

donde $h(\cdot)$ es una función implementada por una red LSTM multicapa con parámetros $\Theta$.

**Naturaleza Autorregresiva del Modelo** La característica distintiva de DeepAR es su naturaleza autorregresiva: en cada paso temporal $t$, la red consume como entrada el valor observado del paso anterior $z_{i,t-1}$ junto con las covariables $x_{i,t}$ y el estado oculto previo $h_{i,t-1}$. Esta estructura permite que el modelo capture dependencias temporales complejas sin necesidad de especificar manualmente órdenes autorregresivos o estructuras de media móvil, como requieren los modelos ARIMA.

Durante el entrenamiento, todos los valores $z_{i,t}$ en el rango de predicción son conocidos y se utilizan directamente en la ecuación (3-76). Durante la predicción, para $t \geq t_0$, los valores futuros son desconocidos, por lo que se reemplazan por muestras $\tilde{z}_{i,t} \sim \ell(\cdot \mid \theta(h_{i,t}, \Theta))$ generadas por el propio modelo, que se retroalimentan para calcular el siguiente estado oculto. Este proceso de *muestreo ancestral* genera trayectorias completas $\tilde{z}_{i,t_0:T}$ que representan realizaciones posibles del futuro.

**Función de Verosimilitud y Modelado de la Incertidumbre** A diferencia de muchos métodos de aprendizaje profundo que se enfocan únicamente en predicciones puntuales, DeepAR está diseñado explícitamente para pronóstico probabilístico. La red no predice directamente el valor futuro $z_{i,t}$, sino los **parámetros de una distribución de probabilidad** $\theta(h_{i,t})$ sobre los valores futuros posibles.

Salinas et al. (2020) proponen dos familias de distribuciones según las características de los datos:

1. **Verosimilitud Gaussiana** para datos de valores reales:

$$\ell_G(z \mid \mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(z-\mu)^2}{2\sigma^2}\right) \tag{3-77}$$

   donde la media $\mu(h_{i,t})$ se obtiene mediante una transformación afín de la salida de la red, y la desviación estándar $\sigma(h_{i,t})$ mediante una transformación afín seguida de una activación softplus para garantizar positividad:

$$\mu(h_{i,t}) = w_\mu^T h_{i,t} + b_\mu, \quad \sigma(h_{i,t}) = \log\bigl(1 + \exp\bigl(w_\sigma^T h_{i,t} + b_\sigma\bigr)\bigr) \tag{3-78}$$

2. **Verosimilitud Binomial Negativa** para datos de conteo:

$$\ell_{NB}(z \mid \mu, \alpha) = \frac{\Gamma(z + 1/\alpha)}{\Gamma(z+1)\Gamma(1/\alpha)} \left(\frac{1}{1+\alpha\mu}\right)^{1/\alpha} \left(\frac{\alpha\mu}{1+\alpha\mu}\right)^{z} \tag{3-79}$$

   donde $\mu \in \mathbb{R}^+$ es la media y $\alpha \in \mathbb{R}^+$ es un parámetro de forma que controla la sobredispersión. En esta parametrización, $\text{Var}[z] = \mu + \mu^2\alpha$, permitiendo modelar varianzas mayores que la media, característica común en datos de demanda intermitente.

La elección de la verosimilitud debe reflejar las propiedades estadísticas de los datos. Para series con valores continuos y errores aproximadamente simétricos, la Gaussiana es apropiada. Para datos de conteo con alta variabilidad o comportamiento intermitente, la Binomial Negativa es preferible.

**Entrenamiento mediante Máxima Verosimilitud** Los parámetros $\Theta$ del modelo se aprenden maximizando la log-verosimilitud sobre todas las series temporales en el conjunto de entrenamiento:

$$\mathcal{L}(\Theta) = \sum_{i=1}^{N} \sum_{t=1}^{T} \log \ell(z_{i,t} \mid \theta(h_{i,t}, \Theta)) \tag{3-80}$$

donde $N$ es el número de series en el conjunto de datos. Esta función objetivo puede optimizarse directamente mediante descenso de gradiente estocástico, ya que $h_{i,t}$ es una función determinística de las entradas y no hay variables latentes que requieran inferencia.

Salinas et al. (2020) destacan una ventaja fundamental de este enfoque: dado que el modelo es completamente observable durante el entrenamiento, no se requieren técnicas de inferencia variacional o métodos de Monte Carlo para aproximar la función objetivo, como sí ocurre en modelos de espacio de estados con variables latentes.

**Manejo de Escalas Heterogéneas: El Problema de la Ley de Potencias** Uno de los desafíos más significativos que aborda DeepAR es la presencia de series temporales con magnitudes que varían en varios órdenes de magnitud. Salinas et al. (2020) documentan que en conjuntos de datos de demanda retail de Amazon, la distribución de velocidades de ventas (ventas promedio por período) sigue aproximadamente una ley de potencias.

Esta distribución implica que un pequeño número de productos tiene ventas extremadamente altas, mientras que la mayoría tiene ventas bajas, con una distribución fuertemente sesgada.

Este fenómeno presenta dos problemas fundamentales:

1. **Problema de rango operativo:** Las no linealidades de la red (funciones de activación como tanh, ReLU) tienen rangos operativos limitados. Sin ajustes, la red debería aprender a escalar las entradas autorregresivas $z_{i,t-1}$ (que pueden variar entre 1 y 10,000) a un rango apropiado en la capa de entrada, y luego invertir este escalamiento en la capa de salida. Este aprendizaje implícito es ineficiente y puede causar problemas de convergencia.
2. **Desbalance en el muestreo:** Si las instancias de entrenamiento se muestrean uniformemente, las series de baja magnitud dominarían el conjunto de datos simplemente por su abundancia numérica, causando que el modelo se ajuste mal a las series de alta magnitud que pueden ser críticas para el negocio.

**Solución 1: Escalamiento Dependiente del Ítem**

DeepAR introduce un mecanismo de escalamiento explícito que normaliza las entradas y salidas autorregresivas por un factor de escala específico de cada serie $\nu_i$. Formalmente, las entradas autorregresivas se transforman como:

$$\tilde{z}_{i,t} = \frac{z_{i,t}}{\nu_i} \tag{3-81}$$

y los parámetros de la verosimilitud dependientes de escala se ajustan correspondientemente. Para la verosimilitud Gaussiana:

$$\mu_{\text{escalado}} = \nu_i \cdot \mu(h_{i,t}), \quad \sigma_{\text{escalado}} = \nu_i \cdot \sigma(h_{i,t}) \tag{3-82}$$

Para la verosimilitud Binomial Negativa:

$$\mu_{\text{escalado}} = \nu_i \cdot \log(1 + \exp(o_\mu)), \quad \alpha_{\text{escalado}} = \frac{\log(1 + \exp(o_\alpha))}{\sqrt{\nu_i}} \tag{3-83}$$

donde $o_\mu, o_\alpha$ son las salidas crudas de la red. El factor de escala típicamente se define como:

$$\nu_i = 1 + \frac{1}{t_0} \sum_{t=1}^{t_0} z_{i,t} \tag{3-84}$$

es decir, el promedio histórico de la serie más uno (para evitar divisiones por cero en series con media cero).

**Solución 2: Muestreo Ponderado por Velocidad**

Para contrarrestar el desbalance introducido por la ley de potencias, DeepAR implementa un esquema de muestreo no uniforme donde la probabilidad de seleccionar una ventana de entrenamiento de la serie $i$ es proporcional a su factor de escala $\nu_i$:

$$P(\text{seleccionar serie } i) \propto \nu_i \tag{3-85}$$

Este muestreo ponderado garantiza que series de alta velocidad sean visitadas con mayor frecuencia durante el entrenamiento, compensando su baja representación numérica y permitiendo que el modelo aprenda patrones específicos de estos ítems críticos.

**Generación de Múltiples Trayectorias y Pronóstico Probabilístico** Una vez entrenado el modelo, la generación de pronósticos probabilísticos se realiza mediante muestreo ancestral. Para cada serie $i$, se generan $B$ trayectorias completas $\{\tilde{z}_{i,t_0:T}^{(b)}\}_{b=1}^{B}$ mediante el siguiente procedimiento:

1. Calcular el estado inicial $h_{i,t_0-1}$ procesando el rango de condicionamiento con los valores observados $z_{i,1:t_0-1}$.
2. Para cada trayectoria $b = 1, \ldots, B$ y cada paso temporal $t = t_0, \ldots, T$:
   a) Calcular los parámetros de la distribución: $\theta_{i,t} = \theta(h_{i,t}, \Theta)$
   b) Muestrear: $\tilde{z}_{i,t}^{(b)} \sim \ell(\cdot \mid \theta_{i,t})$
   c) Actualizar estado: $h_{i,t+1} = h(h_{i,t}, \tilde{z}_{i,t}^{(b)}, x_{i,t+1}, \Theta)$

El conjunto de trayectorias $\{\tilde{z}_{i,t_0:T}^{(b)}\}$ representa una muestra empírica de la distribución predictiva conjunta $Q_\Theta(z_{i,t_0:T} \mid z_{i,1:t_0-1}, x_{i,1:T})$. A partir de estas muestras, se pueden calcular diversos estadísticos de interés:

- **Cuantiles puntuales:** El cuantil $q$-ésimo para el tiempo $t$ se obtiene como el cuantil

empírico de $\{\tilde{z}_{i,t}^{(b)}\}_{b=1}^{B}$

- **Predicción puntual:** La mediana (cuantil 0.5) típicamente se usa como predicción puntual
- **Intervalos de predicción:** Intervalos del $(1-\alpha)\times 100\%$ se construyen mediante $[q_{\alpha/2}, q_{1-\alpha/2}]$
- **Distribuciones agregadas:** Para evaluar la demanda total en un período $[t_0, t_0+S)$, se suma cada trayectoria: $Z_i^{(b)} = \sum_{t=t_0}^{t_0+S-1} \tilde{z}_{i,t}^{(b)}$, y se calculan estadísticos sobre $\{Z_i^{(b)}\}$

Una ventaja crucial de este enfoque es que las trayectorias muestreadas preservan las correlaciones temporales aprendidas por el modelo. Como demuestran Salinas et al. (2020), si se destruyen artificialmente estas correlaciones (mezclando aleatoriamente los valores de cada tiempo entre trayectorias), la calibración de los intervalos de predicción se degrada significativamente, especialmente para horizontes multi-paso.

**Aprendizaje de Patrones Complejos desde los Datos** A diferencia de modelos clásicos donde la estacionalidad, tendencia y crecimiento de la incertidumbre deben especificarse manualmente, DeepAR aprende estos patrones automáticamente desde los datos. Salinas et al. (2020) documentan que el modelo:

- **Aprende estacionalidad heterogénea:** Puede capturar patrones estacionales que varían entre ítems sin requerir especificación manual de períodos estacionales
- **Modela crecimiento no lineal de la incertidumbre:** A diferencia de modelos de espacio de estados que asumen crecimiento lineal de la varianza con el horizonte, DeepAR aprende patrones más complejos. Por ejemplo, en datos de retail, la incertidumbre puede aumentar durante el cuarto trimestre (temporada alta) y luego decrecer en enero
- **Genera pronósticos calibrados:** Los intervalos de predicción tienen cobertura empírica cercana a la cobertura nominal especificada, indicando que el modelo cuantifica adecuadamente la incertidumbre

### De la Teoría a la Práctica

La implementación de DeepAR para series temporales univariadas desarrollada en esta investigación traduce el marco teórico autorregresivo a un sistema predictivo concreto mediante las siguientes adaptaciones específicas al contexto de este estudio.

**Adaptación 1: Simplificación de Covariables** **Teoría Original:** El trabajo de Salinas et al. (2020) asume la disponibilidad de múltiples covariables $x_{i,t}$ tanto dependientes del tiempo (día de la semana, mes del año, indicadores de promociones) como específicas del ítem (categoría de producto, características físicas). Estas covariables se estandarizan y se concatenan con las entradas autorregresivas.

**Adaptación para Series Univariadas:** En el contexto de este estudio, donde el objetivo es evaluar el desempeño de métodos de pronóstico probabilístico en series sintéticas sin información auxiliar, la implementación **no utiliza covariables externas**. La arquitectura se reduce a su forma más pura autorregresiva:

$$h_{i,t} = h(h_{i,t-1}, z_{i,t-1}, \Theta) \tag{3-86}$$

Esta simplificación elimina la dependencia de $x_{i,t}$ en la ecuación (3-76), haciendo que el modelo dependa únicamente de la historia observada de la serie. Matemáticamente, esto equivale a fijar $x_{i,t} = \emptyset$ para todo $t$, o alternativamente, a usar un vector de covariables constante que es absorbido en los parámetros de la red.

**Implicación Técnica:** La capa de entrada de la LSTM tiene dimensión 1 (solo el valor autorregresivo escalado), en lugar de $1 + D_x$ donde $D_x$ sería la dimensionalidad de las covariables. Esta simplificación tiene dos consecuencias:

- **Reducción paramétrica:** Menos parámetros en la capa de entrada aceleran el entrenamiento y reducen el riesgo de sobreajuste
- **Pureza metodológica:** Al no depender de información auxiliar, la comparación entre métodos se centra exclusivamente en la capacidad de cada enfoque para extraer patrones de la historia temporal intrínseca

**Adaptación 2: Construcción de Ventanas de Entrenamiento** **Desafío Práctico:** En el protocolo de simulación de esta tesis, cada configuración genera una única serie temporal

de longitud $n = 252$. A diferencia del contexto original de DeepAR donde se dispone de miles de series relacionadas, aquí el modelo debe aprender exclusivamente de ventanas extraídas de una sola serie.

**Solución Implementada:** Se emplea una estrategia de *windowing* deslizante para generar múltiples instancias de entrenamiento a partir de la serie única. Sea $\{y_1, y_2, \ldots, y_{n_{\text{train}}}\}$ el conjunto de entrenamiento. Se construyen instancias $(X^{(k)}, y^{(k)})$ mediante:

$$X^{(k)} = [y_k, y_{k+1}, \ldots, y_{k+p-1}], \quad y^{(k)} = y_{k+p} \tag{3-87}$$

para $k = 1, \ldots, n_{\text{train}} - p$, donde $p$ es el número de rezagos autorregresivos (hiperparámetro `n_lags`). Esto genera aproximadamente $n_{\text{train}} - p$ instancias de entrenamiento, incrementando artificialmente el tamaño del conjunto de datos.

**Justificación Teórica:** Este procedimiento es válido bajo el supuesto de estacionariedad local, donde se asume que las dependencias temporales son homogéneas en diferentes ventanas temporales. Para los escenarios ARMA y SETAR estacionarios de este estudio, esta suposición se cumple por diseño. Para el escenario ARIMA, la diferenciación previa de la serie garantiza estacionariedad de la serie transformada.

**Adaptación 3: Normalización mediante Z-Score** **Teoría Original:** Salinas et al. (2020) utilizan el escalamiento por $\nu_i$ definido en (3-84) específicamente para manejar la ley de potencias en magnitudes entre series. Este escalamiento no centra los datos (no resta la media).

**Adaptación Implementada:** Para series individuales donde la preocupación principal es la convergencia del entrenamiento y no la comparación entre escalas heterogéneas, se emplea **normalización Z-score completa**:

$$\tilde{z}_t = \frac{z_t - \mu_{\text{train}}}{\sigma_{\text{train}} + \epsilon} \tag{3-88}$$

donde $\mu_{\text{train}}$ y $\sigma_{\text{train}}$ son la media y desviación estándar muestrales del conjunto de entrenamiento, y $\epsilon = 10^{-8}$ es una constante pequeña para estabilidad numérica. Las predicciones se des-normalizan mediante:

$$z_t = \tilde{z}_t \cdot \sigma_{\text{train}} + \mu_{\text{train}} \tag{3-89}$$

**Justificación:** Esta normalización garantiza que las entradas a la LSTM tengan media cero y varianza unitaria, acelerando la convergencia del algoritmo de optimización y previniendo problemas de gradientes explosivos o desvanecientes. Para la verosimilitud Gaussiana empleada en esta implementación, los parámetros predichos $\mu(h_{i,t})$ y $\sigma(h_{i,t})$ están en escala normalizada y se transforman mediante:

$$\mu_{\text{original}} = \mu(h_{i,t}) \cdot \sigma_{\text{train}} + \mu_{\text{train}}, \quad \sigma_{\text{original}} = \sigma(h_{i,t}) \cdot \sigma_{\text{train}} \tag{3-90}$$

**Adaptación 4: Early Stopping con Partición Interna** **Desafío Técnico:** Las redes neuronales recurrentes son susceptibles al sobreajuste, especialmente cuando el número de parámetros es grande relativo al tamaño del conjunto de datos. Sin regularización apropiada, el modelo puede memorizar el ruido en los datos de entrenamiento, degradando el desempeño predictivo.

**Solución Implementada:** Se incorpora un mecanismo de *early stopping* con partición interna del conjunto de entrenamiento. De las $n_{\text{train}} - p$ instancias generadas por windowing, se reserva el 20% final como conjunto de validación interno:

$$n_{\text{val}} = \lfloor 0.2 \times (n_{\text{train}} - p) \rfloor \tag{3-91}$$

Durante el entrenamiento, se monitorea la log-verosimilitud negativa (equivalentemente, la pérdida Gaussian NLL) sobre el conjunto de validación. Se mantiene un contador de paciencia que se incrementa cada época donde la pérdida de validación no mejora. El entrenamiento se detiene si:

$$\text{patience\_counter} \geq \text{patience\_threshold} \tag{3-92}$$

donde el umbral de paciencia es un hiperparámetro (típicamente 5 épocas en esta implementación). Los parámetros del modelo correspondientes a la época con menor pérdida de validación se retienen como modelo final.

**Ventaja sobre Entrenamiento Fijo:** Este esquema adaptativo permite que el entrenamiento continúe mientras se observe mejora, pero previene ciclos innecesarios que solo contribuyen al sobreajuste. Empíricamente, se observa que el entrenamiento típicamente converge entre 15-30 épocas dependiendo de la complejidad de la serie, mucho antes del límite máximo de 30 épocas especificado.

**Adaptación 5: Protocolo de Congelamiento para Rolling Forecast** **Problema Crítico:** En la evaluación rolling window implementada en este estudio, donde se realizan predicciones secuenciales sobre 12 pasos de prueba, un enfoque ingenuo re-entrenaría el modelo en cada paso. Esto introduciría dos problemas fundamentales:

1. **Data leakage:** La re-estimación de parámetros en cada paso permitiría que información futura contamine la evaluación, invalidando las métricas de desempeño
2. **Costo computacional prohibitivo:** Entrenar una red LSTM desde cero es computacionalmente costoso (típicamente 5-15 minutos por serie en hardware estándar). Repetir esto 12 veces por serie haría inviable la experimentación a gran escala

**Solución: Freeze-and-Reuse Protocol**

El protocolo implementado congela completamente el modelo después del entrenamiento inicial:

1. **Fase de Congelamiento:** Sobre el conjunto de entrenamiento ($n_{\text{train}} = 200$ observaciones), se:
   - Estima los parámetros de normalización: $\mu_{\text{frozen}}, \sigma_{\text{frozen}}$
   - Entrena la red LSTM hasta convergencia (early stopping)
   - Almacena los pesos de la red: $\Theta_{\text{frozen}}$
2. **Fase de Evaluación Rolling:** Para cada paso de predicción $t = 1, \ldots, 12$:
   - Normaliza los últimos $p$ valores observados usando $\mu_{\text{frozen}}, \sigma_{\text{frozen}}$
   - Calcula la predicción usando $\Theta_{\text{frozen}}$ (sin re-entrenamiento)
   - Des-normaliza las muestras predictivas

Este protocolo es **estadísticamente válido** porque emula una situación realista donde un analista entrena un modelo una vez con datos históricos disponibles y lo utiliza para hacer pronósticos futuros sin acceso a información posterior. La normalización congelada garantiza que el modelo opere siempre en el mismo espacio transformado que aprendió durante el entrenamiento.

**Implementación Técnica:** La clase `DeepARModel` mantiene flags explícitos:

- `_is_frozen`: Booleano que indica si el modelo ha sido congelado

- `_trained_model`: Referencia a la arquitectura LSTM entrenada
- `_frozen_mean`, `_frozen_std`: Parámetros de normalización congelados

El método `fit_predict` verifica `_is_frozen` al inicio: si es `True`, omite completamente el entrenamiento y procede directamente a generar predicciones con el modelo existente.

### Diferencias Respecto a la Implementación Original de Amazon

La Tabla **3-6** resume las principales diferencias metodológicas entre la implementación original de Salinas et al. (2020) y la adaptación desarrollada en esta tesis.

| **Aspecto** | **DeepAR Original (Salinas et al.)** | **Implementación (Esta Tesis)** |
|---|---|---|
| Contexto de aplicación | Miles de series relacionadas (cross-learning) | Serie temporal única (windowing local) |
| Covariables | Múltiples features temporales e ítem-específicas | Sin covariables externas (solo autorregresivo) |
| Verosimilitud | Gaussiana y Binomial Negativa | Gaussiana únicamente |
| Normalización | Escalamiento por $\nu_i$ (sin centrado) | Z-score completo (centrado y escalado) |
| Muestreo de entrenamiento | Ponderado por velocidad entre series | Uniforme sobre ventanas de serie única |
| Regularización | No especificada en detalle | Early stopping con validación interna |
| Protocolo de evaluación | Modelo único para todas las series | Congelamiento total para rolling forecast |
| Framework | MXNet | PyTorch |
| Objetivo de diseño | Producción a gran escala (500K+ series) | Evaluación experimental controlada |

Table **3-6**: Comparación entre DeepAR Original y DeepAR Adaptado

**Limitaciones de la Adaptación y Desviaciones Teóricas** Es importante reconocer las limitaciones inherentes a esta adaptación de DeepAR para el contexto de series únicas:

1. **Pérdida de cross-learning:** La ventaja fundamental de DeepAR—aprender patrones compartidos entre múltiples series relacionadas—no se explota completamente en este contexto. El modelo aprende únicamente de ventanas de una sola serie, lo que reduce su capacidad de generalización comparado con el escenario multi-serie original.

2. **Tamaño de datos limitado:** Con $n_{\text{train}} = 200$ observaciones, el número de instancias de entrenamiento ($\sim 150-190$ dependiendo de $p$) es relativamente pequeño para una red neuronal profunda. Esto puede limitar la capacidad del modelo de aprender representaciones complejas sin sobreajuste.

3. **Ausencia de regularización de verosimilitud:** Salinas et al. (2020) no detallan técnicas de regularización explícita más allá del dropout entre capas LSTM. En esta implementación, el early stopping actúa como único mecanismo de regularización, lo que puede ser insuficiente para evitar sobreajuste en series altamente ruidosas.

A pesar de estas limitaciones, la implementación mantiene los principios arquitectónicos fundamentales de DeepAR y permite evaluar su desempeño en el marco experimental controlado de este estudio.

#### Parámetros e Hiperparámetros en la Implementación

**Hiperparámetros Arquitectónicos** **hidden_size ($h$):**

- *Tipo:* Entero positivo
- *Valor por defecto:* $h = 20$
- *Rango explorado en optimización:* $\{10, 20, 40\}$
- *Función:* Define la dimensionalidad del estado oculto de cada celda LSTM. Controla la capacidad representacional del modelo.
- *Trade-off:* Valores mayores permiten capturar patrones más complejos pero incrementan el riesgo de sobreajuste y el costo computacional. Para series de longitud moderada ($n \approx 200$), valores en el rango $[10, 40]$ son típicamente suficientes.

**n_lags ($p$):**

- *Tipo:* Entero positivo

- *Valor por defecto:* $p = 5$
- *Rango explorado:* $\{3, 5, 10\}$
- *Función:* Número de valores pasados utilizados como entrada autorregresiva en cada paso temporal
- *Interpretación:* Similar al orden $p$ en modelos AR($p$), determina la memoria del modelo. Valores mayores permiten capturar dependencias de más largo plazo a costa de reducir el número de instancias de entrenamiento.

**num_layers ($L$):**

- *Tipo:* Entero positivo
- *Valor por defecto:* $L = 1$
- *Función:* Número de capas LSTM apiladas
- *Justificación de valor conservador:* Para series univariadas sin covariables complejas, una sola capa LSTM típicamente es suficiente. Múltiples capas incrementan dramáticamente el número de parámetros: $\Theta \propto L \times h^2$, exacerbando el riesgo de sobreajuste en muestras pequeñas.

**dropout ($d$):**

- *Tipo:* Real en $[0, 1)$
- *Valor por defecto:* $d = 0.1$
- *Función:* Tasa de dropout aplicada entre capas LSTM (solo si $L > 1$)
- *Mecanismo:* Durante el entrenamiento, cada conexión entre capas se desactiva con probabilidad $d$, forzando a la red a aprender representaciones robustas

**Hiperparámetros de Optimización lr (learning rate):**

- *Tipo:* Real positivo
- *Valor por defecto:* $\text{lr} = 0.01$
- *Optimizador:* Adam (Kingma and Ba 2015)
- *Función:* Controla el tamaño del paso en la actualización de parámetros. Adam

adapta el learning rate individualmente para cada parámetro basándose en estimaciones de primer y segundo momento del gradiente.

**batch_size ($B$):**

- *Tipo:* Entero positivo
- *Valor por defecto:* $B = 32$
- *Función:* Número de instancias procesadas simultáneamente antes de actualizar parámetros
- *Trade-off:* Batches grandes ($B > 64$) producen estimaciones más estables del gradiente pero requieren más memoria. Batches pequeños ($B < 16$) introducen más ruido estocástico, lo cual puede actuar como regularización implícita pero ralentiza la convergencia.

**epochs ($E$):**

- *Tipo:* Entero positivo
- *Valor por defecto:* $E = 30$
- *Función:* Número máximo de pasadas completas sobre el conjunto de entrenamiento
- *Nota:* Este es un límite superior; el early stopping típicamente detiene el entrenamiento antes de alcanzar este máximo.

**early_stopping_patience ($P$):**

- *Tipo:* Entero positivo
- *Valor por defecto:* $P = 5$
- *Función:* Número de épocas sin mejora en la pérdida de validación antes de detener el entrenamiento
- *Criterio de mejora:* Se considera mejora si $\text{val\_loss}_t < \text{best\_val\_loss} - \epsilon$ donde $\epsilon = 10^{-6}$ es una tolerancia numérica

**Parámetros de Predicción num_samples ($S$):**

- *Tipo:* Entero positivo

- *Valor por defecto:* $S = 1000$
- *Función:* Número de trayectorias muestreadas mediante muestreo ancestral para construir la distribución predictiva
- *Precisión:* Con $S = 1000$, el error estándar de un cuantil empírico es aproximadamente $\sqrt{p(1-p)/1000}$, que es menor al 1.6% incluso en el peor caso ($p = 0.5$)

**random_state:**

- *Tipo:* Entero
- *Valor por defecto:* 42
- *Función:* Semilla para inicialización de pesos de la red y generación de muestras. Garantiza reproducibilidad total de los experimentos

**Proceso de Selección de Hiperparámetros** La optimización de hiperparámetros sigue un protocolo de búsqueda en rejilla conservador que evita sobreajuste a configuraciones específicas:

1. **Grilla de búsqueda:** Se evalúan combinaciones de:
   - `n_lags` $\in \{3, 5, 10\}$
   - `hidden_size` $\in \{10, 20, 40\}$
   - `num_layers` $\in \{1, 2\}$ (opcional, típicamente fijo en 1)
2. **Métrica de selección:** CRPS promedio sobre el conjunto de calibración (40 observaciones posteriores al entrenamiento)
3. **Congelamiento:** La configuración óptima identificada se congela para toda la fase de evaluación rolling

**Configuración Típica Resultante** Para la mayoría de las series en los escenarios de simulación, la configuración óptima converge a:

$$\{\texttt{n\_lags} = 5, \texttt{hidden\_size} = 20, \texttt{num\_layers} = 1, \texttt{dropout} = 0.1\} \tag{3-93}$$

Esta configuración balanceada resulta en aproximadamente $20 \times (20 + 5 + 1) \times 4 = 2080$

parámetros para la capa LSTM (considerando las cuatro gates: input, forget, cell, output), más los parámetros de las capas de salida $\mu$ y $\sigma$, totalizando aproximadamente 2200 parámetros. Con $\sim$ 150 instancias de entrenamiento, la relación parámetros/datos es aproximadamente 15 : 1, lo cual es manejable con regularización apropiada.

### 3.4.8 Autoregressive Exponentially-weighted Polynomial Distribution (AREPD)

#### Propuesta Teórica

El modelo *Autoregressive Exponentially-weighted Polynomial Distribution* (AREPD) representa una contribución metodológica original de esta investigación, desarrollada como una extensión híbrida que combina elementos de predicción conformal ponderada con expansión polinomial de características autorregresivas. A diferencia de los métodos puramente conformales como LSPM o LSPMW que utilizan regresión lineal en el espacio original de rezagos, AREPD introduce **transformaciones no lineales polinomiales** de las entradas autorregresivas, permitiendo capturar relaciones no lineales entre valores pasados y futuros sin recurrir a arquitecturas de aprendizaje profundo.

**Motivación: Limitaciones de la Linealidad en Modelos Conformales** Los métodos conformales basados en mínimos cuadrados, como el LSPM de Vovk, Gammerman, and Shafer (2022), asumen implícitamente que la relación entre valores pasados $Y_{t-1}, Y_{t-2}, \ldots, Y_{t-p}$ y el valor futuro $Y_t$ puede aproximarse adecuadamente mediante una función lineal:

$$\mathbb{E}[Y_t \mid \mathcal{F}_{t-1}] \approx \beta_0 + \sum_{j=1}^{p} \beta_j Y_{t-j} \tag{3-94}$$

Esta suposición es válida para procesos lineales estacionarios como ARMA, donde la estructura autorregresiva es inherentemente lineal por construcción. Sin embargo, para procesos no lineales como SETAR, donde la dinámica cambia según el estado del sistema, o para series con efectos de amplificación (donde valores grandes en el pasado producen valores desproporcionadamente grandes en el futuro), la aproximación lineal puede ser inadecuada.

AREPD aborda esta limitación mediante la expansión polinomial del espacio de características. Para cada rezago $Y_{t-j}$, se generan términos polinomiales $Y_{t-j}, Y_{t-j}^2, \ldots, Y_{t-j}^d$ donde $d$ es el grado polinomial. Esto permite que el modelo capture relaciones cuadráticas, cúbicas, o de orden superior entre el pasado y el futuro, manteniendo la estructura de regresión lineal pero en un espacio de características enriquecido.

**Fundamento Matemático: Regresión Ridge Ponderada con Expansión Polinomial** Sea $\{Y_t\}_{t=1}^n$ una serie temporal observada. Para un número de rezagos $p$ y un grado polinomial $d$, AREPD construye una matriz de diseño expandida $\mathbf{X} \in \mathbb{R}^{(n-p)\times(1+pd)}$ donde cada fila corresponde a una instancia temporal:

$$\mathbf{X}_i = \left[1, Y_i, Y_i^2, \ldots, Y_i^d, Y_{i+1}, Y_{i+1}^2, \ldots, Y_{i+1}^d, \ldots, Y_{i+p-1}, Y_{i+p-1}^2, \ldots, Y_{i+p-1}^d\right] \tag{3-95}$$

para $i = 1, \ldots, n-p$. El vector objetivo correspondiente es:

$$\mathbf{y} = [Y_{p+1}, Y_{p+2}, \ldots, Y_n]^T \in \mathbb{R}^{n-p} \tag{3-96}$$

La inclusión del término constante (1) en la primera posición de cada fila permite que el modelo capture un nivel base no cero, equivalente a un intercepto en regresión lineal estándar.

**Ponderación Exponencial Temporal** Inspirado en el esquema de decaimiento temporal de Barber et al. (2023) para predicción conformal adaptativa, AREPD asigna pesos exponencialmente decrecientes a las observaciones históricas. Para la observación en el tiempo $t = p + i$, el peso asociado es:

$$w_i = \rho^{n-p-i}, \quad i = 1, \ldots, n-p \tag{3-97}$$

donde $\rho \in (0, 1)$ es el **parámetro de decaimiento**. Los pesos se normalizan para sumar uno:

$$\tilde{w}_i = \frac{w_i}{\sum_{j=1}^{n-p} w_j} \tag{3-98}$$

Este esquema implica que la observación más reciente (tiempo $n$) recibe el peso máximo $\tilde{w}_{n-p} \propto \rho^0 = 1$, mientras que observaciones más antiguas reciben pesos progresivamente menores. El parámetro $\rho$ controla la velocidad del olvido:

- $\rho \approx 1$: Memoria larga, todas las observaciones tienen pesos similares
- $\rho \approx 0.8$: Memoria corta, el pasado reciente domina completamente
- $\rho = 0.95$: Valor intermedio típico que balancea estabilidad y adaptabilidad

La vida media efectiva de la información bajo este esquema es:

$$\tau_{1/2} = \frac{\log(2)}{\log(1/\rho)} \tag{3-99}$$

Por ejemplo, con $\rho = 0.95$, $\tau_{1/2} \approx 13.5$ observaciones, lo que significa que una observación de hace 14 pasos temporales contribuye con aproximadamente la mitad del peso de una observación actual.

**Estimación mediante Regresión Ridge** Los coeficientes del modelo se estiman resolviendo el problema de regresión Ridge ponderada:

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta}} \left\{ \sum_{i=1}^{n-p} \tilde{w}_i \left( y_i - \mathbf{X}_i^T \boldsymbol{\beta} \right)^2 + \lambda \|\boldsymbol{\beta}\|_2^2 \right\} \tag{3-100}$$

donde $\lambda > 0$ es el parámetro de regularización Ridge (también denotado $\alpha$ en la implementación) que penaliza la norma $L_2$ de los coeficientes. La inclusión de esta regularización es crítica por dos razones:

1. **Estabilidad numérica:** La expansión polinomial de orden alto ($d \geq 2$) puede generar matrices de diseño casi singulares debido a la alta correlación entre términos como $Y_{t-1}$ y $Y_{t-1}^2$. La penalización Ridge garantiza que $\mathbf{X}^T\mathbf{W}\mathbf{X} + \lambda\mathbf{I}$ sea bien condicionada.
2. **Prevención de sobreajuste:** Con $p \cdot d$ características (excluyendo el intercepto), el modelo tiene alta capacidad expresiva. Sin regularización, podría ajustarse al ruido en el conjunto de entrenamiento, especialmente cuando $n - p$ es relativamente pequeño.

La solución del problema (3-100) tiene forma cerrada:

$$\hat{\boldsymbol{\beta}} = \left(\mathbf{X}^T\mathbf{W}\mathbf{X} + \lambda\mathbf{I}\right)^{-1}\mathbf{X}^T\mathbf{W}\mathbf{y} \tag{3-101}$$

donde $\mathbf{W} = \text{diag}(\tilde{w}_1, \ldots, \tilde{w}_{n-p})$ es la matriz diagonal de pesos. Esta formulación es computacionalmente eficiente y puede resolverse mediante descomposición de Cholesky o SVD para máxima estabilidad numérica.

**Generación de Distribuciones Predictivas** Una vez estimados los coeficientes $\hat{\boldsymbol{\beta}}$, AREPD genera una distribución predictiva mediante un enfoque **histórico-empírico** que difiere fundamentalmente de los métodos conformales puros. Para predecir el siguiente valor $Y_{n+1}$, el modelo:

1. Construye el vector de características expandido para todos los puntos históricos:

$$\mathbf{X}_{\text{hist}}[i,:] = \text{PolyExpand}(Y_i, Y_{i+1}, \ldots, Y_{i+p-1}), \quad i = 1, \ldots, n-p \tag{3-102}$$

2. Calcula las predicciones puntuales históricas:

$$\hat{Y}_{\text{hist}} = \mathbf{X}_{\text{hist}}\hat{\boldsymbol{\beta}} \tag{3-103}$$

3. Transforma estas predicciones a la escala original:

$$\hat{Y}_{\text{hist}}^{\text{original}} = \hat{Y}_{\text{hist}} \cdot \sigma_{\text{train}} + \mu_{\text{train}} \tag{3-104}$$

4. Utiliza este conjunto $\{\hat{Y}_{\text{hist}}^{\text{original}}\}$ como **muestra empírica de la distribución predictiva**.

Este enfoque es conceptualmente distinto de la predicción conformal estándar. Mientras que métodos como LSPM calculan scores conformales $C_i = h(x_{\text{test}}) + (y_i - h(x_i))$ que ajustan la predicción puntual mediante residuos de calibración, AREPD construye una distribución empírica directamente a partir de las predicciones históricas del modelo ajustado. La justificación subyacente es que, si el modelo captura adecuadamente la estructura de dependencia temporal, las predicciones sobre datos históricos deberían ser representativas de la distribución predictiva futura bajo el supuesto de estacionariedad local.

**Propiedades Teóricas y Garantías de Cobertura** A diferencia de los métodos conformales con garantías de cobertura finita demostradas formalmente, AREPD **no posee garantías teóricas de cobertura bajo intercambiabilidad**. La razón fundamental es que la distribución predictiva no se construye mediante el framework conformal estándar, sino mediante un mecanismo histórico-empírico que asume:

$$\hat{Y}_t \mid \mathcal{F}_{t-1} \overset{d}{\approx} \hat{Y}_s \mid \mathcal{F}_{s-1} \quad \forall t, s \in \{p+1, \ldots, n\} \tag{3-105}$$

Esta suposición de distribución estacionaria de las predicciones puede violarse en presencia de:

- **No estacionariedad fuerte:** Si la serie exhibe tendencias o cambios estructurales, las predicciones históricas pueden tener distribuciones sistemáticamente diferentes de las predicciones futuras.
- **Heteroscedasticidad condicional:** Si la varianza de los errores cambia con el tiempo (como en procesos GARCH), el rango de las predicciones históricas puede no reflejar adecuadamente la incertidumbre futura.
- **Eventos raros no observados:** Si el período de entrenamiento no contiene eventos extremos, la distribución empírica subestimará las colas de la verdadera distribución predictiva.

A pesar de estas limitaciones teóricas, AREPD puede exhibir buen desempeño empírico en escenarios donde:

1. La no linealidad del proceso es suave y capturables mediante polinomios de bajo orden ($d \leq 3$)
2. La estacionariedad se mantiene aproximadamente en el horizonte de evaluación
3. El conjunto de entrenamiento es suficientemente largo para que las predicciones históricas cubran el rango de valores futuros plausibles

**Comparación Conceptual con Métodos Relacionados** La Tabla **3-7** posiciona AREPD en el espacio de métodos autorregresivos para pronóstico probabilístico.

AREPD ocupa un nicho intermedio entre métodos conformales lineales simples (LSPM/LSPMW) y arquitecturas de aprendizaje profundo complejas (DeepAR). Su capacidad de capturar

| **Método** | **Espacio de características** | **Ponderación temporal** | **Distribución predictiva** |
|---|---|---|---|
| LSPM | Lineal (rezagos) | Uniforme | Conformal (scores ajustados) |
| LSPMW | Lineal (rezagos) | Exponencial ($\rho$) | Conformal ponderada |
| AREPD | Polinomial (hasta grado $d$) | Exponencial ($\rho$) | Histórico-empírica |
| DeepAR | No lineal (LSTM) | Uniforme (en entrenamiento) | Muestreo ancestral |
| Sieve Bootstrap | Lineal (orden $p$ creciente) | Uniforme | Bootstrap de residuos |

Table **3-7**: Comparación conceptual de AREPD con métodos relacionados

no linealidades mediante expansión polinomial lo hace potencialmente más expresivo que LSPM para procesos no lineales, pero sin el costo computacional y los requisitos de datos de DeepAR.

### De la Teoría a la Práctica

La implementación de AREPD desarrollada para esta investigación traduce el marco teórico en un sistema predictivo concreto mediante las siguientes decisiones de diseño y adaptaciones específicas.

**Adaptación 1: Normalización Z-Score Pre-Expansión** **Desafío Numérico:** La expansión polinomial es extremadamente sensible a la escala de las variables de entrada. Si una serie tiene valores en el rango $[100, 200]$, los términos cuadráticos estarán en el rango $[10^4, 4 \times 10^4]$ y los términos cúbicos en $[10^6, 8 \times 10^6]$. Esta explosión de escalas causa:

- **Inestabilidad numérica:** La matriz $\mathbf{X}^T\mathbf{W}\mathbf{X}$ puede tener número de condición extremadamente alto ($> 10^{10}$), haciendo la inversión matricial numéricamente inestable.
- **Dominancia de términos de alto orden:** Los coeficientes asociados a términos cúbicos dominarían la predicción simplemente por su escala, no por su importancia predictiva real.

**Solución Implementada:** Se aplica normalización Z-score **antes** de la expansión polinomial:

$$\tilde{Y}_t = \frac{Y_t - \mu_{\text{train}}}{\sigma_{\text{train}} + \epsilon} \tag{3-106}$$

donde $\epsilon = 10^{-8}$ previene división por cero. La expansión polinomial se aplica entonces a los valores normalizados $\tilde{Y}_t$, garantizando que todos los términos estén aproximadamente en el rango $[-3, 3]$ bajo normalidad. Las predicciones se des-normalizan al final:

$$Y_t^{\text{pred}} = \tilde{Y}_t^{\text{pred}} \cdot \sigma_{\text{train}} + \mu_{\text{train}} \tag{3-107}$$

Esta transformación es matemáticamente válida porque preserva las relaciones polinomiales:

$$f(\tilde{Y}_{t-1}, \tilde{Y}_{t-2}, \ldots) = g(Y_{t-1}, Y_{t-2}, \ldots) \text{ si } f \text{ y } g \text{ son polinomios} \tag{3-108}$$

aunque con coeficientes reescalados apropiadamente.

**Adaptación 2: Construcción Eficiente de la Matriz de Diseño** **Implementación Vectorizada:** La construcción de la matriz de diseño en (3-95) se implementa mediante operaciones vectorizadas de NumPy para eficiencia computacional. El pseudocódigo es:

```
X_list = [np.ones((n-p, 1))]  # Término constante

for lag in range(p):
    lagged_values = Y[lag : lag + (n-p)]
    for degree in range(1, d + 1):
        X_list.append(lagged_values ** degree)

X = np.hstack(X_list)
```

Esta construcción genera una matriz con estructura de bloques:

$$\mathbf{X} = \begin{bmatrix} \mathbf{1} & \mathbf{Y}_0 & \mathbf{Y}_0^2 & \cdots & \mathbf{Y}_0^d & \mathbf{Y}_1 & \mathbf{Y}_1^2 & \cdots & \mathbf{Y}_{p-1}^d \end{bmatrix} \tag{3-109}$$

donde $\mathbf{Y}_{\text{lag}}$ denota el vector columna de valores con rezago específico.

**Adaptación 3: Uso de scikit-learn Ridge con Sample Weights** **Ventaja de Biblioteca Establecida:** En lugar de implementar manualmente la solución de (3-101), se utiliza la clase `Ridge` de scikit-learn con el argumento `sample_weight`. Esta elección proporciona:

- **Estabilidad numérica:** scikit-learn utiliza descomposición SVD o Cholesky optimizada dependiendo del condicionamiento de la matriz
- **Validación automática:** Manejo robusto de casos extremos (matrices singulares, pesos negativos, etc.)
- **Eficiencia:** Código optimizado en C/Cython para operaciones matriciales

La implementación es directa:

```
from sklearn.linear_model import Ridge

model = Ridge(alpha=lambda, fit_intercept=False)
model.fit(X, y, sample_weight=weights)
predictions = model.predict(X)
```

El argumento `fit_intercept=False` es crucial porque el término constante ya está incluido explícitamente como primera columna de $\mathbf{X}$.

**Adaptación 4: Protocolo de Congelamiento Consistente** Siguiendo la filosofía unificada implementada en todos los modelos de este estudio, AREPD adopta un protocolo de congelamiento completo que fija:

1. **Parámetros de normalización:** $\mu_{\text{frozen}}, \sigma_{\text{frozen}}$ calculados sobre el conjunto de entrenamiento
2. **Coeficientes del modelo:** $\hat{\boldsymbol{\beta}}_{\text{frozen}}$ estimados mediante (3-101) sobre datos de entrenamiento
3. **Distribución predictiva base:** El conjunto de predicciones históricas transformadas que constituye la distribución empírica

Durante la fase de evaluación rolling, para cada nuevo punto de prueba:

- Se normalizan los últimos $p$ valores observados usando $\mu_{\text{frozen}}, \sigma_{\text{frozen}}$
- Se expande polinomialmente hasta grado $d$
- Se aplica $\hat{\boldsymbol{\beta}}_{\text{frozen}}$ (sin re-estimación)
- Se retorna la distribución histórica des-normalizada

Este protocolo garantiza ausencia de data leakage y permite comparación equitativa con otros métodos bajo condiciones idénticas.

**Adaptación 5: Manejo de Casos Degenerados Series Muy Cortas:** Si $n < 2p$, no hay suficientes observaciones para construir la matriz de diseño. El modelo degrada a un predictor constante:

$$\hat{Y}_{n+1} = \mu_{\text{frozen}} \tag{3-110}$$

con distribución predictiva degenerada (punto de masa en la media).

**Matriz de Diseño Singular:** Si después de la regularización Ridge la matriz sigue siendo numéricamente singular (número de condición $> 10^{12}$), la predicción falla y se retorna un fallback similar al caso anterior.

### Parámetros e Hiperparámetros en la Implementación

#### Hiperparámetros Estructurales n_lags ($p$):

- *Tipo:* Entero positivo
- *Valor por defecto:* $p = 5$
- *Rango explorado:* $\{3, 5, 7, 10\}$
- *Función:* Número de rezagos incluidos en la matriz de diseño
- *Impacto dimensional:* Con grado polinomial $d$, la dimensión del espacio de características es $1 + p \cdot d$

- *Trade-off:* Valores mayores permiten capturar dependencias de largo plazo, pero reducen el tamaño del conjunto de entrenamiento ($n - p$ instancias disponibles) e incrementan el riesgo de sobreajuste

**poly_degree ($d$):**

- *Tipo:* Entero $\geq 1$
- *Valor por defecto:* $d = 2$ (términos cuadráticos)
- *Rango explorado:* $\{1, 2, 3\}$
- *Función:* Grado máximo de la expansión polinomial
- *Interpretación:*
  - $d = 1$: Modelo puramente lineal, equivalente a regresión autorregresiva Ridge estándar
  - $d = 2$: Incluye términos cuadráticos, puede capturar efectos de amplificación moderados
  - $d = 3$: Incluye términos cúbicos, alta expresividad pero riesgo significativo de sobreajuste
- *Número de parámetros:* # params $= 1 + p \cdot d$. Para $(p, d) = (5, 2)$: 11 parámetros; para $(p, d) = (10, 3)$: 31 parámetros.

**Hiperparámetros de Ponderación y Regularización rho ($\rho$):**

- *Tipo:* Real en $(0, 1)$
- *Valor por defecto:* $\rho = 0.95$
- *Rango explorado:* $\{0.90, 0.95, 0.98\}$
- *Función:* Parámetro de decaimiento exponencial en el esquema de ponderación temporal definido en (3-97)
- *Vida media efectiva:*
  - $\rho = 0.90$: $\tau_{1/2} \approx 6.6$ observaciones
  - $\rho = 0.95$: $\tau_{1/2} \approx 13.5$ observaciones

  - $\rho = 0.98$: $\tau_{1/2} \approx 34.3$ observaciones

- *Impacto:* Valores altos ($\rho > 0.95$) aproximan ponderación uniforme, útil para series estacionarias estables. Valores bajos ($\rho < 0.90$) permiten adaptación rápida a cambios, apropiado para series con drift o cambios estructurales graduales.

**alpha ($\lambda$):**

- *Tipo:* Real positivo
- *Valor por defecto:* $\lambda = 0.1$
- *Función:* Parámetro de regularización Ridge en (3-100)
- *Efecto:* Controla el trade-off sesgo-varianza:
  - $\lambda \to 0$: Aproxima mínimos cuadrados no regularizados, máxima flexibilidad pero riesgo de sobreajuste
  - $\lambda \to \infty$: Contrae coeficientes hacia cero, produciendo predictor casi constante
  - $\lambda = 0.1$: Valor intermedio que proporciona regularización suave
- *Selección:* En esta implementación, $\lambda$ se mantiene fijo y no se optimiza. Una extensión futura podría incluir validación cruzada para selección automática de $\lambda$.

**Configuración Típica Resultante** Para la mayoría de las series en los escenarios de simulación, la configuración óptima converge a:

$$\{\texttt{n\_lags} = 5, \texttt{poly\_degree} = 2, \texttt{rho} = 0.95, \texttt{alpha} = 0.1\} \tag{3-111}$$

Esta configuración genera 11 parámetros a estimar:

- 1 término constante
- 5 términos lineales $(Y_{t-1}, Y_{t-2}, \ldots, Y_{t-5})$
- 5 términos cuadráticos $(Y_{t-1}^2, Y_{t-2}^2, \ldots, Y_{t-5}^2)$

Con $n_{\text{train}} = 200$ observaciones, se dispone de aproximadamente 195 instancias de entrenamiento, resultando en una relación parámetros/datos de aproximadamente $1 : 18$, que es favorable para prevenir sobreajuste con la regularización Ridge activa.

**Proceso de Selección de Hiperparámetros** La optimización de hiperparámetros sigue un protocolo de búsqueda en rejilla sobre el espacio:

$$\mathcal{H} = \{(p, d, \rho) : p \in \{3, 5, 7\}, d \in \{1, 2, 3\}, \rho \in \{0.90, 0.95, 0.98\}\} \tag{3-112}$$

Para cada combinación:

1. Se entrena el modelo Ridge ponderado sobre el conjunto de entrenamiento
2. Se calculan predicciones sobre el conjunto de calibración (40 observaciones)
3. Se evalúa el CRPS promedio como métrica de selección
4. Se retiene la configuración con menor CRPS

La configuración óptima se congela para toda la evaluación rolling posterior.

**Consideraciones sobre Interpretabilidad** A diferencia de arquitecturas de caja negra como DeepAR, AREPD mantiene cierta interpretabilidad debido a su estructura de regresión lineal en el espacio polinomial expandido. Los coeficientes $\hat{\boldsymbol{\beta}}$ pueden inspeccionarse para identificar:

- **Rezagos más influyentes:** Rezagos con coeficientes lineales de gran magnitud ($|\beta_j| >$ umbral)
- **Presencia de no linealidades:** Coeficientes cuadráticos o cúbicos significativamente diferentes de cero indican relaciones no lineales
- **Dirección de dependencias:** Signo positivo/negativo de coeficientes lineales indica correlación positiva/negativa con rezagos específicos

Esta transparencia puede ser valiosa en aplicaciones donde la comprensión del modelo es importante, como pronóstico de demanda o planificación operativa.

### Posicionamiento Metodológico y Limitaciones

#### Ventajas Potenciales de AREPD

1. **Flexibilidad no lineal:** Puede capturar relaciones polinomiales sin arquitecturas complejas de aprendizaje profundo. Para procesos como SETAR con cambios de

régimen suaves, la expansión cuadrática puede aproximar razonablemente las transiciones entre regímenes.

2. **Eficiencia computacional:** El entrenamiento se reduce a resolver un sistema lineal, significativamente más rápido que optimizar redes neuronales. Típicamente requiere menos de 1 segundo por serie en hardware estándar.

3. **Adaptabilidad temporal:** El parámetro $\rho$ permite ajustar la memoria del modelo sin cambiar la arquitectura, útil para series con diferentes grados de estacionariedad local.

4. **Regularización explícita:** El parámetro Ridge $\lambda$ proporciona control directo sobre el trade-off sesgo-varianza, más transparente que técnicas como dropout en redes neuronales.

5. **Interpretabilidad parcial:** Los coeficientes pueden inspeccionarse para entender qué patrones temporales captura el modelo, algo imposible en arquitecturas de caja negra.

#### Limitaciones Fundamentales

1. **Ausencia de garantías conformales:** El mecanismo de construcción de la distribución predictiva no está fundamentado en teoría conformal, por lo que no hereda las garantías de cobertura finita de métodos como LSPM. La cobertura empírica depende críticamente de la validez del supuesto de estacionariedad de las predicciones históricas.

2. **Maldición de la dimensionalidad polinomial:** Para grados altos ($d \geq 4$) y múltiples rezagos ($p > 10$), el número de parámetros crece como $O(pd)$, haciendo el modelo propenso a sobreajuste incluso con regularización Ridge. Para $(p, d) = (15, 4)$, se requerirían estimar 61 parámetros, lo cual es excesivo para series de longitud moderada.

3. **Inestabilidad en extrapolación:** Los polinomios de orden alto son notoriamente inestables fuera del rango de los datos de entrenamiento. Si la serie de prueba contiene valores extremos no observados durante el entrenamiento, los términos cúbicos o de orden superior pueden producir predicciones arbitrariamente grandes o pequeñas.

4. **Incapacidad para cambios estructurales abruptos:** A diferencia del MCPS que

puede adaptarse localmente mediante bins, o DeepAR que aprende patrones complejos, AREPD asume que la estructura polinomial aprendida es globalmente válida. Cambios de régimen abruptos (como en SETAR con umbrales nítidos) pueden no ser bien capturados por aproximaciones polinomiales suaves.

5. **Dependencia de normalización:** El rendimiento del método depende críticamente de la normalización Z-score. Series con outliers extremos pueden sesgar $\mu_{\text{train}}$ y $\sigma_{\text{train}}$, degradando la estabilidad numérica de la expansión polinomial.

6. **Distribución predictiva histórica:** El uso de predicciones históricas como distribución predictiva asume implícitamente que los errores del modelo son homocedásticos (varianza constante) y estacionarios. Esto puede ser inadecuado para series con volatilidad cambiante o errores dependientes del nivel.

**Casos de Uso Recomendados** Basándose en las características teóricas, AREPD es potencialmente más apropiado para:

- Series temporales con no linealidades suaves capturables mediante polinomios de bajo orden ($d \leq 3$)
- Escenarios donde se requiere interpretabilidad básica de la estructura de dependencia
- Situaciones donde el costo computacional es una restricción severa
- Procesos estacionarios o localmente estacionarios sin cambios estructurales abruptos

Y menos apropiado para:

- Series con cambios de régimen nítidos o discontinuidades
- Procesos con heterocedasticidad condicional fuerte (volatilidad cambiante)
- Situaciones donde garantías formales de cobertura son requeridas
- Series con valores extremos frecuentes que violan la aproximación polinomial

**Innovación Metodológica** La contribución de AREPD no reside en sus garantías teóricas (que son limitadas), sino en demostrar empíricamente si la combinación de:

$$\text{Expansión Polinomial} + \text{Ponderación Temporal} + \text{Regularización Ridge} \tag{3-113}$$

puede proporcionar un balance efectivo entre expresividad no lineal y simplicidad computacional en el contexto de pronóstico probabilístico. Los resultados experimentales del Capítulo **??** permitirán cuantificar si esta estrategia híbrida ofrece ventajas sobre métodos puramente lineales (LSPM/LSPMW) o si la complejidad adicional no se traduce en mejoras de desempeño medidas por ECRPS.

En caso de que AREPD exhiba desempeño competitivo en escenarios no lineales como SETAR, esto proporcionaría evidencia de que aproximaciones polinomiales simples pueden ser suficientes para ciertas clases de no linealidades en series temporales, sin necesidad de recurrir a métodos conformales localmente adaptativos (MCPS) o arquitecturas de aprendizaje profundo (DeepAR). Por el contrario, si su desempeño es inferior, esto validaría la necesidad de enfoques más sofisticados con fundamentación teórica más robusta.

### 3.4.9 Ensemble Conformalized Quantile Regression con LSTM (EnCQR-LSTM)

#### Fundamento Teórico

El modelo *Ensemble Conformalized Quantile Regression* (EnCQR) con arquitectura LSTM representa una aproximación híbrida que combina regresión cuantílica mediante redes neuronales recurrentes, aprendizaje en ensamble, y predicción conformal para construir intervalos de predicción (PIs) probabilísticos con garantías de cobertura para series temporales heterocedásticas (Jensen, Bianchi, and Anfinsen 2022).

**Motivación: Limitaciones de Métodos Puros** Los métodos basados únicamente en regresión cuantílica (QR), aunque pueden generar intervalos adaptativos que ajustan su amplitud a la variabilidad local de los datos, carecen de garantías formales de cobertura. En la práctica, los PIs generados por QR tienden a ser excesivamente confiados (demasiado estrechos), resultando en coberturas empíricas significativamente inferiores al nivel de confianza nominal $(1 - \alpha)$.

Por otro lado, los métodos de predicción conformal (CP) estándar garantizan cobertura marginal válida bajo el supuesto de intercambiabilidad de los datos. Sin embargo, CP tradicional construye intervalos de amplitud constante o levemente variable, lo cual es inadecuado para series temporales con heterocedasticidad, donde la incertidumbre varía

considerablemente a lo largo del tiempo. Para datos con variabilidad cambiante, estos intervalos constantes resultan:

- **Excesivamente conservadores** en períodos de baja volatilidad, generando intervalos innecesariamente amplios que proporcionan poca información útil
- **Potencialmente insuficientes** en períodos de alta volatilidad, donde la amplitud fija puede no capturar adecuadamente la incertidumbre real

EnCQR aborda estas limitaciones mediante una síntesis metodológica que hereda las fortalezas de ambos enfoques: la adaptabilidad local de QR y las garantías de cobertura de CP.

**Arquitectura de Ensamble y Partición de Datos** EnCQR construye un ensamble homogéneo de $B$ aprendices base, cada uno entrenado sobre subconjuntos **disjuntos** de los datos de entrenamiento. Formalmente, dado un conjunto de entrenamiento $\{(x_i, y_i)\}_{i=1}^{T}$, se particiona en $B$ subconjuntos consecutivos no solapados:

$$S_b = \{(x_i, y_i) : i \in [(b-1)T_b + 1, bT_b]\}, \quad b = 1, \ldots, B \tag{3-114}$$

donde $T_b = \lfloor T/B \rfloor$ es la longitud de cada subconjunto. Esta partición disjunta es fundamental para la construcción de residuos fuera de muestra válidos, ya que garantiza que cada observación está excluida de al menos un modelo del ensamble.

Cada subconjunto $S_b$ se utiliza para entrenar un modelo LSTM de regresión cuantílica que estima simultáneamente múltiples funciones cuantílicas condicionales (CQFs):

$$\hat{q}_{\tau}^{(b)}(x) = \mathrm{LSTM}_b(x; \theta_b), \quad \tau \in \{\tau_{\mathrm{lo}}, 0.50, \tau_{\mathrm{hi}}\} \tag{3-115}$$

donde $\theta_b$ representa los parámetros del $b$-ésimo modelo LSTM, y los niveles cuantílicos $\tau_{\mathrm{lo}}$ y $\tau_{\mathrm{hi}}$ definen los límites inferior y superior del intervalo nominal. Típicamente, para un nivel de confianza $(1-\alpha)$, se establecen valores iniciales como $\tau_{\mathrm{lo}} = \alpha/2$ y $\tau_{\mathrm{hi}} = 1 - \alpha/2$.

**Regresión Cuantílica mediante Pinball Loss** El entrenamiento de cada modelo LSTM se realiza minimizando la función de pérdida pinball agregada sobre todos los cuantiles objetivo:

$$\mathcal{L}_{\text{pinball}}(\theta_b) = \frac{1}{|S_b| \cdot |\mathcal{T}|} \sum_{(x_i,y_i)\in S_b} \sum_{\tau\in\mathcal{T}} \rho_\tau(y_i - \hat{q}_\tau^{(b)}(x_i)) \tag{3-116}$$

donde $\mathcal{T} = \{\tau_{\text{lo}}, 0.50, \tau_{\text{hi}}\}$ es el conjunto de cuantiles objetivo, y $\rho_\tau(\cdot)$ es la función de pérdida pinball definida como:

$$\rho_\tau(u) = \begin{cases} \tau \cdot u, & \text{si } u \geq 0 \\ (\tau - 1) \cdot u, & \text{si } u < 0 \end{cases} \tag{3-117}$$

Esta función penaliza asimétricamente los errores de predicción: para cuantiles superiores ($\tau > 0.5$), las subestimaciones reciben mayor penalización, mientras que para cuantiles inferiores ($\tau < 0.5$), las sobreestimaciones son más penalizadas. Esta asimetría es crucial para que el modelo aprenda a estimar correctamente los cuantiles condicionales de la distribución de $Y|X$.

**Predicción Leave-One-Out del Ensamble** Una vez entrenados los $B$ modelos, EnCQR construye predicciones leave-one-out (LOO) para cada observación de entrenamiento. Para una observación $i$ en el subconjunto $S_b$, se agregan las predicciones de todos los modelos entrenados **sin** incluir esa observación:

$$\hat{q}_\tau^{(-i)}(x_i) = \phi\left(\{\hat{q}_\tau^{(b')}(x_i) : S_{b'} \text{ tal que } i \notin S_{b'}\}\right) \tag{3-118}$$

donde $\phi(\cdot)$ es una función de agregación, típicamente la media aritmética. Este procedimiento LOO es esencial para aplicar predicción conformal a series temporales, ya que reemplaza el requisito de intercambiabilidad por un esquema de validación cruzada que genera residuos genuinamente fuera de muestra.

**Scores de Conformidad Asimétricos** EnCQR introduce scores de conformidad **asimétricos** que cuantifican separadamente el error de cobertura en las colas inferior y superior de la distribución:

$$\begin{aligned} E_i^{\text{lo}} &= \hat{q}_{\tau_{\text{lo}}}^{(-i)}(x_i) - y_i \\ E_i^{\text{hi}} &= y_i - \hat{q}_{\tau_{\text{hi}}}^{(-i)}(x_i) \end{aligned} \tag{3-119}$$

para $i = 1, \ldots, T$. La motivación para esta formulación asimétrica es que las distribuciones de los errores para los cuantiles inferior y superior pueden tener formas diferentes, especialmente en presencia de asimetría en los errores del modelo. Si se utilizara un score simétrico único (como en CP estándar), un error de cobertura asimétrico podría distribuirse incorrectamente entre las colas, resultando en cobertura inferior al nivel nominal.

Los scores $\{E_i^{\text{lo}}\}_{i=1}^T$ y $\{E_i^{\text{hi}}\}_{i=1}^T$ forman dos distribuciones empíricas que cuantifican cuánto debe expandirse cada límite del intervalo para garantizar la cobertura deseada.

**Conformalización de Intervalos de Predicción** Para una nueva observación $x_{T+1}$, el ensamble completo genera predicciones agregadas:

$$\begin{aligned} \hat{q}_{\tau_{\text{lo}}}(x_{T+1}) &= \phi\left(\{\hat{q}_{\tau_{\text{lo}}}^{(b)}(x_{T+1})\}_{b=1}^B\right) \\ \hat{q}_{\tau_{\text{hi}}}(x_{T+1}) &= \phi\left(\{\hat{q}_{\tau_{\text{hi}}}^{(b)}(x_{T+1})\}_{b=1}^B\right) \end{aligned} \tag{3-120}$$

Estos cuantiles agregados se conformalizan ajustando su amplitud mediante los cuantiles $(1-\alpha)$ de las distribuciones de scores:

$$\hat{C}_\alpha(x_{T+1}) = \left[\hat{q}_{\tau_{\text{lo}}}(x_{T+1}) - \omega^{\text{lo}}, \hat{q}_{\tau_{\text{hi}}}(x_{T+1}) + \omega^{\text{hi}}\right] \tag{3-121}$$

donde:

$$\begin{aligned} \omega^{\text{lo}} &= Q_{1-\alpha}(\{E_i^{\text{lo}}\}_{i=1}^T) \\ \omega^{\text{hi}} &= Q_{1-\alpha}(\{E_i^{\text{hi}}\}_{i=1}^T) \end{aligned} \tag{3-122}$$

y $Q_{1-\alpha}(\cdot)$ denota el cuantil empírico $(1-\alpha)$ de una colección de valores.

Este mecanismo de conformalización garantiza que, bajo el supuesto de que los errores residuales $\{E_i^{\text{lo}}, E_i^{\text{hi}}\}$ son aproximadamente estacionarios y fuertemente mezclantes (condiciones más débiles que intercambiabilidad), el intervalo $\hat{C}_\alpha(x_{T+1})$ satisface:

$$\mathbb{P}\{Y_{T+1} \in \hat{C}_\alpha(X_{T+1})\} \geq 1 - \alpha + O(1/T) \tag{3-123}$$

con alta probabilidad para muestras grandes.

**Adaptación Temporal mediante Ventana Deslizante** Para series temporales no estacionarias, EnCQR incorpora un mecanismo de actualización mediante ventana deslizante. Cada $s$ nuevas observaciones (donde $s$ corresponde típicamente a un ciclo estacional, e.g., 24 horas para datos horarios), los scores de conformidad se actualizan:

1. Se calculan los nuevos residuos LOO para las últimas $s$ observaciones
2. Se eliminan los $s$ residuos más antiguos de las colecciones $\{E_i^{\text{lo}}\}$ y $\{E_i^{\text{hi}}\}$
3. Se incorporan los nuevos residuos, manteniendo constante el tamaño total de las colecciones

Este protocolo de actualización permite que los factores de conformalización $\omega^{\text{lo}}$ y $\omega^{\text{hi}}$ se adapten a cambios graduales en la variabilidad de la serie, manteniendo la cobertura válida sin necesidad de reentrenar los modelos LSTM del ensamble.

**Propiedades Teóricas** EnCQR posee las siguientes propiedades formales:

1. **Cobertura marginal aproximadamente válida:** Bajo el supuesto de que el proceso de error es estacionario y fuertemente mezclante, EnCQR garantiza que la cobertura marginal converge al nivel nominal $(1-\alpha)$ cuando $T \to \infty$.
2. **Adaptabilidad heterocedástica:** A diferencia de CP estándar, la amplitud del intervalo en (3-121) varía localmente con $x_{T+1}$ a través de las predicciones cuantílicas $\hat{q}_\tau(x_{T+1})$, permitiendo intervalos más estrechos en regiones de baja variabilidad.
3. **Distribución-libre:** No se asumen formas paramétricas específicas para la distribución de $Y|X$. La única suposición estructural es la estacionariedad débil del proceso de error.
4. **Robustez ante especificación incorrecta:** Incluso si el modelo LSTM subyacente está mal especificado, la conformalización garantiza cobertura válida, aunque los intervalos pueden ser más amplios de lo necesario.

### Arquitectura LSTM para Regresión Cuantílica

**Estructura de Red Neuronal Recurrente** La arquitectura LSTM utilizada en EnCQR consiste en una secuencia de capas recurrentes LSTM seguidas de una capa densa completamente conectada que produce las estimaciones cuantílicas. Formalmente, para una

ventana temporal de entrada $\mathbf{x}_t = [y_{t-N_x}, y_{t-N_x+1}, \ldots, y_{t-1}] \in \mathbb{R}^{N_x}$, donde $N_x$ es la longitud de la ventana, la red procesa:

$$\begin{aligned} \mathbf{h}_t^{(1)} &= \mathrm{LSTM}^{(1)}(\mathbf{x}_t, \mathbf{h}_{t-1}^{(1)}) \\ \mathbf{h}_t^{(2)} &= \mathrm{LSTM}^{(2)}(\mathbf{h}_t^{(1)}, \mathbf{h}_{t-1}^{(2)}) \\ &\vdots \\ \mathbf{h}_t^{(L)} &= \mathrm{LSTM}^{(L)}(\mathbf{h}_t^{(L-1)}, \mathbf{h}_{t-1}^{(L)}) \\ \hat{\mathbf{q}}_t &= \mathbf{W}_{\text{out}}\mathbf{h}_t^{(L)} + \mathbf{b}_{\text{out}} \end{aligned} \tag{3-124}$$

donde $L$ es el número de capas LSTM apiladas, $\mathbf{h}_t^{(\ell)}$ representa el estado oculto de la $\ell$-ésima capa en el tiempo $t$, y $\hat{\mathbf{q}}_t = [\hat{q}_{\tau_{\text{lo}}}, \hat{q}_{0.50}, \hat{q}_{\tau_{\text{hi}}}]^T \in \mathbb{R}^3$ es el vector de cuantiles estimados.

**Mecanismo de Estados LSTM** Cada celda LSTM mantiene dos estados: el estado oculto $\mathbf{h}_t$ y el estado de celda $\mathbf{c}_t$, actualizados mediante:

$$\begin{aligned} \mathbf{f}_t &= \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \quad \text{(forget gate)} \\ \mathbf{i}_t &= \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \quad \text{(input gate)} \\ \tilde{\mathbf{c}}_t &= \tanh(\mathbf{W}_c[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_c) \quad \text{(candidate values)} \\ \mathbf{c}_t &= \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t \quad \text{(cell state update)} \\ \mathbf{o}_t &= \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \quad \text{(output gate)} \\ \mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \quad \text{(hidden state)} \end{aligned} \tag{3-125}$$

donde $\sigma(\cdot)$ es la función sigmoide, $\odot$ denota multiplicación elemento a elemento, y $[\cdot, \cdot]$ representa concatenación. Esta arquitectura permite que el modelo capture dependencias temporales de largo plazo, esencial para series con memoria extendida.

**Normalización y Regularización** La arquitectura incluye:

- **Normalización MinMax:** Los datos de entrada se escalan al rango $[0, 1]$ mediante:

$$\tilde{y}_t = \frac{y_t - y_{\min}}{y_{\max} - y_{\min}} \tag{3-126}$$

  preservando $\mu_{\text{train}}, \sigma_{\text{train}}$ para des-normalización posterior.

- **Regularización L2:** Se aplica penalización Ridge a todos los pesos de la red:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{pinball}} + \lambda_2 \sum_{\ell=1}^{L} \|\mathbf{W}^{(\ell)}\|_F^2 \tag{3-127}$$

  donde $\|\cdot\|_F$ denota la norma de Frobenius.

- **Dropout:** Con probabilidad $p_{\text{drop}} = 0.1$ después de cada capa LSTM (excepto la última) para prevenir sobreajuste.

### Generación de Distribuciones Predictivas

**Ajuste de Distribución Skew-Normal** Una vez obtenidos los cuantiles conformalizados $\hat{q}_\tau^{\text{conf}} = [\hat{q}_{0.01}, \hat{q}_{0.05}, \ldots, \hat{q}_{0.99}]$ para múltiples niveles $\tau$, EnCQR construye una distribución predictiva continua ajustando una distribución **Skew-Normal** paramétrica a estos cuantiles.

La elección de la distribución Skew-Normal está motivada por:

1. **Flexibilidad asimétrica:** Puede representar distribuciones simétricas (cuando el parámetro de asimetría $\alpha = 0$, reduciendo a Normal) o asimétricas, común en series temporales reales.
2. **Unimodalidad garantizada:** A diferencia de interpolaciones lineales entre cuantiles que pueden producir distribuciones bimodales artificiales, la Skew-Normal es unimodal por construcción.
3. **Parsimonia:** Requiere solo tres parámetros $(\mu, \sigma, \alpha)$ correspondientes a localización, escala y asimetría.

La función de densidad de la distribución Skew-Normal es:

$$f(x; \mu, \sigma, \alpha) = \frac{2}{\sigma} \phi\left(\frac{x-\mu}{\sigma}\right) \Phi\left(\alpha \frac{x-\mu}{\sigma}\right) \tag{3-128}$$

donde $\phi(\cdot)$ y $\Phi(\cdot)$ son la función de densidad y distribución acumulada de la Normal estándar, respectivamente.

**Estimación de Parámetros** Los parámetros $(\mu, \sigma, \alpha)$ se estiman minimizando la distancia cuadrática entre los cuantiles empíricos conformalizados y los cuantiles teóricos de la Skew-Normal:

$$(\hat{\mu}, \hat{\sigma}, \hat{\alpha}) = \arg\min_{(\mu,\sigma,\alpha)} \sum_{i=1}^{|\mathcal{T}|} \left( \hat{q}_{\tau_i}^{\text{conf}} - F_{\text{SN}}^{-1}(\tau_i; \mu, \sigma, \alpha) \right)^2 \tag{3-129}$$

donde $F_{\text{SN}}^{-1}(\tau; \mu, \sigma, \alpha)$ es la función cuantílica de la Skew-Normal. Esta optimización se resuelve mediante el método L-BFGS-B con restricciones en el dominio:

$$\begin{aligned} &\mu \in [\min(\hat{q}_{\tau}^{\text{conf}}) - \text{IQR}, \max(\hat{q}_{\tau}^{\text{conf}}) + \text{IQR}] \\ &\sigma \in [\text{IQR}/10, 3 \cdot \text{IQR}], \quad \sigma > 0 \\ &\alpha \in [-5, 5] \end{aligned} \tag{3-130}$$

donde $\text{IQR} = \hat{q}_{0.75}^{\text{conf}} - \hat{q}_{0.25}^{\text{conf}}$ es el rango intercuartil.

**Inicialización Robusta** Los valores iniciales para la optimización se establecen como:

$$\begin{aligned} &\mu_0 = \hat{q}_{0.50}^{\text{conf}} \quad \text{(mediana empírica)} \\ &\sigma_0 = \text{IQR}/1.35 \quad \text{(aproximación Normal estándar)} \\ &\alpha_0 = 0 \quad \text{(iniciar simétrico)} \end{aligned} \tag{3-131}$$

Si la optimización no converge, se utiliza un *fallback* a distribución Normal pura ($\alpha = 0$) con parámetros estimados mediante momentos muestrales.

**Muestreo de la Distribución Predictiva** Una vez estimada la distribución Skew-Normal $\text{SN}(\hat{\mu}, \hat{\sigma}, \hat{\alpha})$, se generan $M = 1000$ muestras mediante:

$$\tilde{Y}_{T+1}^{(m)} \sim \text{SN}(\hat{\mu}, \hat{\sigma}, \hat{\alpha}), \quad m = 1, \ldots, M \tag{3-132}$$

Este conjunto de muestras $\{\tilde{Y}_{T+1}^{(m)}\}_{m=1}^{M}$ constituye una representación empírica de la distribución predictiva completa, permitiendo:

- Cálculo de cualquier cuantil predictivo mediante interpolación

- Estimación de momentos (media, varianza, asimetría, curtosis)
- Evaluación de métricas probabilísticas como CRPS
- Visualización de la densidad predictiva mediante histogramas o KDE

#### Parámetros e Hiperparámetros

**Hiperparámetros de Arquitectura LSTM** **n_lags ($N_x$):**

- *Tipo:* Entero positivo
- *Valor por defecto:* $N_x = 20$
- *Rango explorado:* $\{10, 20, 30, 40\}$
- *Función:* Longitud de la ventana temporal de entrada
- *Impacto:* Valores mayores permiten capturar dependencias de más largo plazo, pero reducen el número efectivo de muestras de entrenamiento a $T - N_x$ y aumentan la complejidad computacional

**units ($N_u$):**

- *Tipo:* Entero positivo
- *Valor por defecto:* $N_u = 32$
- *Rango explorado:* $\{16, 32, 64, 128\}$
- *Función:* Número de unidades (dimensión del estado oculto) en cada capa LSTM
- *Capacidad expresiva:* Determina la cantidad de memoria y capacidad de representación. Valores altos incrementan expresividad pero riesgo de sobreajuste

**n_layers ($L$):**

- *Tipo:* Entero positivo
- *Valor por defecto:* $L = 2$
- *Rango explorado:* $\{1, 2, 3\}$
- *Función:* Número de capas LSTM apiladas

- *Interpretación:* Capas adicionales permiten representaciones jerárquicas a diferentes escalas temporales, pero incrementan significativamente el número de parámetros

**Hiperparámetros de Entrenamiento** **lr ($\eta$):**

- *Tipo:* Real positivo
- *Valor por defecto:* $\eta = 0.005$
- *Rango explorado:* $[10^{-4}, 10^{-2}]$ en escala logarítmica
- *Función:* Tasa de aprendizaje del optimizador Adam
- *Trade-off:* Valores altos aceleran convergencia pero pueden causar inestabilidad; valores bajos requieren más épocas pero convergen más suavemente

**batch_size ($B_{\text{train}}$):**

- *Tipo:* Entero positivo
- *Valor por defecto:* $B_{\text{train}} = 16$
- *Rango explorado:* $\{8, 16, 32, 64\}$
- *Función:* Tamaño de lotes para entrenamiento por descenso de gradiente estocástico
- *Impacto:* Lotes pequeños proporcionan actualizaciones ruidosas que pueden ayudar a escapar mínimos locales; lotes grandes dan estimaciones más estables pero requieren más memoria

**epochs:**

- *Tipo:* Entero positivo
- *Valor por defecto:* 20 (con early stopping)
- *Función:* Número máximo de pasadas sobre el conjunto de entrenamiento
- *Early stopping:* El entrenamiento se detiene si la pérdida de validación no mejora durante 50 épocas consecutivas, previniendo sobreajuste

**Hiperparámetros de Ensamble y Conformalización** **B (número de modelos):**

- *Tipo:* Entero positivo

- *Valor por defecto:* $B = 3$
- *Función:* Número de aprendices en el ensamble
- *Trade-off:* Valores mayores incrementan diversidad y robustez, pero reducen el tamaño de cada subconjunto de entrenamiento $T_b = T/B$ y aumentan el costo computacional linealmente
- *Justificación:* Para series de longitud $T \approx 200 - 500$, $B = 3$ proporciona balance adecuado, generando subconjuntos de $\approx 65 - 165$ observaciones

**alpha ($\alpha$):**

- *Tipo:* Real en $(0, 1)$
- *Valor por defecto:* $\alpha = 0.05$ (nivel de confianza 95%)
- *Función:* Nivel de error nominal para los intervalos de predicción
- *Interpretación:* Determina el cuantil $(1 - \alpha)$ de los scores de conformidad utilizados en la conformalización

**num_samples ($M$):**

- *Tipo:* Entero positivo
- *Valor por defecto:* $M = 1000$
- *Función:* Número de muestras generadas de la distribución Skew-Normal ajustada
- *Impacto:* Determina la resolución de la distribución predictiva empírica. Valores $M \geq 1000$ proporcionan representaciones suficientemente finas para la mayoría de métricas

**Regularización Lambda L2 ($\lambda_2$):**

- *Rango explorado:* $[10^{-5}, 10^{-1}]$ en escala logarítmica
- *Función:* Penalización Ridge aplicada a todos los pesos de la red
- *Efecto:* Controla el trade-off sesgo-varianza, previniendo que los pesos crezcan excesivamente y causando sobreajuste

**Dropout:**

- *Valor fijo:*
- *Valor fijo:* $p_{\text{drop}} = 0.1$
- *Función:* Probabilidad de desactivación aleatoria de unidades LSTM durante el entrenamiento
- *Efecto:* Regularización implícita que fuerza al modelo a aprender representaciones redundantes y robustas

**Hiperparámetros de Cuantiles Cuantiles objetivo:**

- *Conjunto completo:* $\mathcal{T} = \{0.01, 0.05, 0.10, 0.25, 0.50, 0.75, 0.90, 0.95, 0.99\}$
- *Función:* Niveles cuantílicos estimados simultáneamente por cada LSTM
- *Justificación:* Esta granularidad captura adecuadamente la forma de la distribución predictiva, incluyendo colas extremas (0.01, 0.99) para detectar valores atípicos, cuartiles (0.25, 0.75) para caracterizar dispersión central, y mediana (0.50) para la tendencia central

**Niveles conformalizados optimizables:**

- $\tau_{\text{lo}} \in [0.01, 0.20]$ y $\tau_{\text{hi}} \in [0.70, 0.99]$
- Estos niveles nominales del modelo LSTM subyacente pueden ajustarse como hiperparámetros adicionales para compensar sesgos sistemáticos antes de la conformalización
- La conformalización posterior garantiza cobertura válida independientemente de estos valores

**Configuración Típica y Número de Parámetros** Para la configuración estándar con $N_x = 20$, $N_u = 32$, $L = 2$ capas LSTM, el número total de parámetros entrenables por modelo es aproximadamente:

$$\begin{aligned} \#\text{ params} &= \sum_{\ell=1}^{L} \left[ 4N_u(N_u + d_{\text{in}}^{(\ell)} + 1) \right] + (N_u \cdot |\mathcal{T}| + |\mathcal{T}|) \\ &\approx 4 \cdot 32 \cdot (32 + 1 + 1) + 4 \cdot 32 \cdot (32 + 32 + 1) + 32 \cdot 9 + 9 \\ &\approx 4{,}352 + 8{,}320 + 288 + 9 \approx 12{,}969 \text{ parámetros} \end{aligned} \tag{3-133}$$

donde $d_{\text{in}}^{(1)} = 1$ para la primera capa (entrada univariada) y $d_{\text{in}}^{(\ell)} = N_u$ para capas superiores. El factor 4 corresponde a las cuatro matrices de pesos de cada celda LSTM (forget, input, candidate, output gates).

Con $B = 3$ modelos en el ensamble, el sistema completo requiere entrenar $\approx$ 39,000 parámetros totales, lo cual es manejable para series de longitud $T \geq 200$.

### Protocolo de Congelamiento y Evaluación

**Fase de Entrenamiento y Calibración** El protocolo de congelamiento de EnCQR-LSTM sigue una estructura de dos etapas:

1. **Entrenamiento del ensamble:** Los $B = 3$ modelos LSTM se entrenan sobre sus respectivos subconjuntos disjuntos $S_1, S_2, S_3$ del conjunto de entrenamiento, minimizando la pérdida pinball (3-116) mediante el optimizador Adam con los hiperparámetros seleccionados.
2. **Cálculo de scores de conformidad:** Se generan predicciones LOO para todas las observaciones de entrenamiento según (3-118), y se calculan los scores asimétricos (3-119). Las colecciones $\{E_i^{\text{lo}}\}_{i=1}^{T}$ y $\{E_i^{\text{hi}}\}_{i=1}^{T}$ se almacenan para uso posterior.
3. **Congelamiento completo:** Se fijan permanentemente:
   - Los pesos $\{\theta_b\}_{b=1}^{B}$ de los $B$ modelos LSTM
   - Los parámetros de normalización $(\mu_{\text{train}}, \sigma_{\text{train}})$ o $(y_{\min}, y_{\max})$
   - Las distribuciones empíricas de scores conformales
   - El tamaño de ventana de actualización $s = 24$

**Evaluación Rolling Window** Durante la fase de evaluación, para cada nuevo punto de prueba $t = T+1, T+2, \ldots, T+T'$:

1. Se extrae la ventana de entrada $\mathbf{x}_t = [y_{t-N_x}, \ldots, y_{t-1}]$
2. Se normaliza usando los parámetros congelados:

$$\tilde{\mathbf{x}}_t = \frac{\mathbf{x}_t - \mu_{\text{train}}}{\sigma_{\text{train}} + \epsilon} \tag{3-134}$$

3. Cada modelo del ensamble genera predicciones cuantílicas:

$$\hat{\mathbf{q}}_t^{(b)} = \text{LSTM}_b(\tilde{\mathbf{x}}_t; \theta_b^{\text{frozen}}), \quad b = 1, \ldots, B \tag{3-135}$$

4. Se agregan mediante media aritmética:

$$\hat{\mathbf{q}}_t = \frac{1}{B} \sum_{b=1}^{B} \hat{\mathbf{q}}_t^{(b)} \tag{3-136}$$

5. Se conformalizan los cuantiles:

$$\begin{aligned} \hat{q}_{\tau_i}^{\text{conf}}(t) &= \hat{q}_{\tau_i}(t) - \omega^{\text{lo}} \quad \text{si } \tau_i \leq 0.5 \\ \hat{q}_{\tau_i}^{\text{conf}}(t) &= \hat{q}_{\tau_i}(t) + \omega^{\text{hi}} \quad \text{si } \tau_i > 0.5 \end{aligned} \tag{3-137}$$

   donde $\omega^{\text{lo}} = Q_{1-\alpha}(\{E_i^{\text{lo}}\})$ y $\omega^{\text{hi}} = Q_{1-\alpha}(\{E_i^{\text{hi}}\})$ utilizan las distribuciones de scores actuales.

6. Se des-normalizan los cuantiles conformalizados:

$$\hat{q}_{\tau_i}^{\text{conf, orig}}(t) = \hat{q}_{\tau_i}^{\text{conf}}(t) \cdot \sigma_{\text{train}} + \mu_{\text{train}} \tag{3-138}$$

7. Se ajusta la distribución Skew-Normal mediante (3-129) y se generan $M = 1000$ muestras.

8. **Actualización condicional:** Si $(t - T) \mod s = 0$:
   - Se calculan los nuevos residuos LOO para las observaciones $\{t - s, \ldots, t - 1\}$ usando los modelos congelados
   - Se eliminan los $s$ residuos más antiguos de $\{E_i^{\text{lo}}\}$ y $\{E_i^{\text{hi}}\}$
   - Se añaden los nuevos residuos, manteniendo el tamaño constante
   - Se recalculan $\omega^{\text{lo}}$ y $\omega^{\text{hi}}$ para las predicciones subsiguientes

Este protocolo garantiza:

- **Ausencia de data leakage:** Los modelos nunca observan datos de prueba durante el entrenamiento

- **Evaluación justa:** Todos los métodos operan bajo condiciones idénticas de congelamiento
- **Adaptabilidad controlada:** La actualización de scores permite adaptación a cambios graduales sin reentrenamiento completo

### Posicionamiento Metodológico

**Comparación con Métodos Relacionados** La Tabla 3-8 posiciona EnCQR-LSTM en el espacio de métodos probabilísticos para series temporales.

| Método | Intervalos adaptativos | Cobertura válida | Arquitectura | Complejidad computacional |
|---|---|---|---|---|
| QRNN-LSTM | Sí | No | LSTM con pinball loss | Media (entrenamiento NN) |
| CP estándar | No | Sí | Agnóstico | Baja (post-procesamiento) |
| CQR | Sí | Sí (i.i.d.) | Agnóstico + QR | Media |
| EnbPI | No | Sí (TS) | Ensamble + CP | Media-Alta |
| EnCQR-LSTM | Sí | Sí (TS) | Ensamble LSTM + QR + CP | Alta (B × entrenamiento NN) |
| DeepAR | Sí | No | LSTM autorregresivo | Alta |
| AREPD | Sí | No | Regresión polinomial Ridge | Baja |

Table **3-8**: Comparación conceptual de EnCQR-LSTM con métodos relacionados

### Ventajas Distintivas

1. **Síntesis metodológica óptima:** EnCQR combina tres paradigmas complementarios:
   - *Regresión cuantílica:* Permite intervalos adaptativos que varían con $x$ según la variabilidad local estimada

- *Predicción conformal:* Proporciona garantías de cobertura marginal válida sin suposiciones distribucionales paramétricas
- *Aprendizaje en ensamble:* Mejora robustez mediante agregación de modelos diversos, reduciendo varianza y sensibilidad a inicializaciones

2. **Aplicabilidad a series temporales heterocedásticas:** A diferencia de CP estándar (intervalos constantes) o EnbPI (intervalos simétricos), EnCQR genera intervalos que se expanden y contraen naturalmente según la volatilidad local, crucial para series con variabilidad cambiante.
3. **Robustez ante especificación incorrecta:** Incluso si el modelo LSTM subyacente captura imperfectamente la dinámica temporal, la conformalización garantiza cobertura válida. Los intervalos serán más amplios de lo necesario, pero mantendrán la propiedad de cobertura $(1 - \alpha)$.
4. **Distribución predictiva completa:** El ajuste Skew-Normal permite generar muestras de toda la distribución predictiva, no solo intervalos puntuales. Esto facilita:
   - Estimación de métricas probabilísticas completas (CRPS, Log-Score)
   - Análisis de asimetría y colas de la distribución
   - Toma de decisiones basada en riesgo mediante percentiles arbitrarios
5. **Actualización adaptativa sin reentrenamiento:** El mecanismo de ventana deslizante permite que los factores de conformalización se adapten a cambios graduales en $T$ pasos sin el costoso proceso de reentrenar los $B$ modelos LSTM.
6. **Escalabilidad a arquitecturas complejas:** El framework EnCQR es agnóstico a la arquitectura de red neuronal específica. Puede aplicarse sobre:
   - LSTMs (implementado aquí)
   - Redes convolucionales temporales (TCN)
   - Transformers para series temporales
   - Modelos híbridos CNN-LSTM

#### Limitaciones Fundamentales

1. **Alto costo computacional:** Entrenar $B$ modelos LSTM completos es computacionalmente intensivo:
   - Tiempo de entrenamiento: $O(B \cdot \text{épocas} \cdot T_b \cdot N_u^2)$ donde $T_b = T/B$
   - Para series largas ($T > 10{,}000$) y $B \geq 5$, el tiempo total puede ser prohibitivo sin aceleración GPU
2. **Requisitos de datos:** La partición en $B$ subconjuntos disjuntos reduce efectivamente el tamaño de datos para cada modelo:
   - Cada LSTM se entrena con solo $T/B$ observaciones
   - Para series cortas ($T < 100$), la fragmentación puede ser excesiva, degradando la calidad de los aprendices individuales
   - Existe un trade-off fundamental: $B$ grande mejora diversidad del ensamble pero empobrece datos por modelo
3. **Cobertura marginal, no condicional:** EnCQR garantiza cobertura promedio sobre todo el conjunto de prueba, pero no garantiza cobertura para cada punto individual $x$ específico:

$$\mathbb{P}\left\{\frac{1}{T'}\sum_{t=T+1}^{T+T'} \mathbb{1}\{Y_t \in \hat{C}_\alpha(X_t)\} \geq 1-\alpha\right\} \approx 1 \tag{3-139}$$

   pero no necesariamente $\mathbb{P}\{Y_t \in \hat{C}_\alpha(X_t) \mid X_t = x\} \geq 1-\alpha$ para todo $x$.
4. **Supuestos de estacionariedad débil:** La validez de las garantías de cobertura requiere que el proceso de error sea aproximadamente estacionario y mezclante. Para series con:
   - Cambios estructurales abruptos (quiebres de régimen)
   - Tendencias determinísticas fuertes no removidas
   - Estacionalidad no capturada por el modelo

   la cobertura puede degradarse significativamente.
5. **Ventana de actualización $s$ fija:** El parámetro $s$ debe especificarse a priori basándose en el conocimiento del dominio (e.g., $s = 24$ para ciclos diarios). Una

selección inadecuada puede resultar en:

- $s$ muy pequeño: Actualizaciones frecuentes con alta varianza en $\omega^{\text{lo}}, \omega^{\text{hi}}$, causando intervalos erráticos
- $s$ muy grande: Adaptación lenta a cambios en volatilidad, intervalos potencialmente mal calibrados

6. **Sensibilidad al ajuste Skew-Normal:** Si los cuantiles conformalizados exhiben patrones multimodales o altamente irregulares (posible con muestras pequeñas), el ajuste unimodal Skew-Normal puede ser inadecuado, subestimando la verdadera complejidad de la distribución predictiva.
7. **Ausencia de interpretabilidad:** Las LSTMs son modelos de caja negra. A diferencia de AREPD (coeficientes polinomiales interpretables) o ARMA (coeficientes autorregresivos claros), es difícil extraer conocimiento sobre qué patrones temporales específicos captura el modelo.

**Casos de Uso Recomendados** EnCQR-LSTM es particularmente apropiado para:

- **Series heterocedásticas con patrones complejos:** Donde la volatilidad cambia significativamente (e.g., demanda eléctrica con ciclos diarios/semanales fuertes, series financieras de alta frecuencia).
- **Escenarios donde garantías de cobertura son críticas:** Aplicaciones en planificación energética, gestión de inventarios, o predicción de demanda donde subestimar incertidumbre tiene consecuencias costosas.
- **Datos suficientemente largos:** Series con $T \geq 500$ observaciones permiten particiones en $B = 3$ subconjuntos con $\approx 165$ observaciones cada uno, suficiente para entrenar LSTMs efectivos.
- **Disponibilidad de recursos computacionales:** Entorno con GPUs disponibles para entrenamiento paralelo de los $B$ modelos.
- **Necesidad de distribuciones predictivas completas:** Cuando se requiere más que intervalos puntuales, como análisis de riesgo en colas, optimización estocástica, o toma de decisiones bajo incertidumbre.

Y menos apropiado para:

- Series muy cortas ($T < 200$) donde la partición en $B$ subconjuntos resulta en datos insuficientes por modelo
- Aplicaciones de tiempo real con restricciones computacionales severas donde el overhead de $B$ inferencias es prohibitivo
- Series altamente no estacionarias con cambios estructurales frecuentes que violan el supuesto de estacionariedad débil
- Escenarios donde interpretabilidad del modelo es un requisito regulatorio o de negocio
- Procesos lineales simples donde métodos más parsimoniosos (ARMA, LSPM) serían igualmente efectivos

**Contribución Metodológica en el Contexto del Estudio** EnCQR-LSTM representa el enfoque más sofisticado metodológicamente entre los modelos evaluados en esta investigación, combinando tres paradigmas de frontera en aprendizaje estadístico. Su inclusión permite:

1. **Evaluar el valor de la complejidad arquitectónica:** Cuantificar si la inversión en complejidad computacional y conceptual se traduce en mejoras medibles de ECRPS versus métodos más simples.
2. **Establecer un límite superior de desempeño:** Como uno de los métodos más avanzados disponibles para pronóstico probabilístico de series temporales, EnCQR-LSTM proporciona un benchmark de referencia contra el cual métodos más parsimoniosos pueden compararse.
3. **Validar garantías teóricas en la práctica:** Verificar empíricamente si las garantías formales de cobertura se materializan en los escenarios de simulación considerados (ARMA, SETAR, GARCH).
4. **Caracterizar el trade-off complejidad-desempeño:** Determinar en qué tipos de procesos (lineales, no lineales, heterocedásticos) la complejidad adicional de EnCQR justifica su costo versus alternativas como AREPD o MCPS.

Los resultados empíricos del Capítulo **??** revelarán si EnCQR-LSTM logra su objetivo de sintetizar adaptabilidad heterocedástica con cobertura válida, y a qué costo relativo en términos de complejidad y recursos computacionales.

# Bibliografía


Arrieta Prieto, Mario Enrique (2017). "Evaluation of the Sieve Bootstrap's performance in comparison with the classic approach for forecasting purposes in time series analysis". In: *XXVII Simposio Internacional de Estad'istica / 5th International Workshop on Applied Statistics*. Poster Presentation. Medell'ın, Colombia.

Barber, Rina Foygel et al. (2023). *Conformal Prediction Beyond Exchangeability*. arXiv preprint arXiv:2202.13415v5. Version 5. Accessed on January 5, 2026. arXiv: 2202.13415 [stat.ME].

Bollerslev, Tim (1986). "Generalized Autoregressive Conditional Heteroskedasticity". In: *Journal of Econometrics* 31.3, pp. 307–327. DOI: 10.1016/0304-4076(86)90063-1.

Boström, Henrik (2022). "crepes: a Python Package for Generating Conformal Regressors and Predictive Systems". In: *Conformal and Probabilistic Prediction with Applications*. Vol. 179. PMLR, pp. 24–41.

Boström, Henrik, Ulf Johansson, and Tuve Löfström (2021). "Mondrian Conformal Predictive Distributions". In: *Proceedings of Machine Learning Research* 152, pp. 24–38.

Box, George E. P. and Gwilym M. Jenkins (1968). *Some Recent Advances in Forecasting and Control*. Vol. 17. 2. Wiley, pp. 91–109.

Bühlmann, Peter (1997). "Sieve Bootstrap for Time Series". In: *Bernoulli* 3.2, pp. 123–148. DOI: 10.2307/3318434.

Chan, Kung-Sik and Howell Tong (1985). "Testing for threshold autoregression". In: *The Annals of Statistics* 13.3, pp. 1121–1142.

Chen, Pu and Willi Semmler (2023). "Stability in Threshold VAR Models". In: *Studies in Nonlinear Dynamics and Econometrics*. DOI: 10.1515/snde-2022-0099.

Chen, Tianqi and Carlos Guestrin (2016). "XGBoost: A Scalable Tree Boosting System". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, pp. 785–794.

Coroneo, Laura and Fabrizio Iacone (2020). "Comparing Predictive Accuracy in Small Samples Using Fixed-Smoothing Asymptotics". In: *Journal of Applied Econometrics* 35.3, pp. 391–409. DOI: 10.1002/jae.2756.

Diebold, Francis X. and Roberto S. Mariano (1995). "Comparing Predictive Accuracy". In: *Journal of Business & Economic Statistics* 13.3, pp. 253–263. DOI: 10.1080/07350015.1995.10524599.

Engle, Robert F. (1982). "Autoregressive Conditional Heteroscedasticity with Estimates of the Variance of United Kingdom Inflation". In: *Econometrica* 50.4, pp. 987–1007. DOI: 10.2307/1912773.

Giacomini, Raffaella and Halbert White (2006). "Tests of Conditional Predictive Ability". In: *Econometrica* 74.6, pp. 1545–1578. DOI: 10.1111/j.1468-0262.2006.00718.x.

Gneiting, Tilmann, Fadoua Balabdaoui, and Adrian E. Raftery (2007). "Probabilistic Forecasts, Calibration and Sharpness". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 69.2, pp. 243–268. DOI: 10.1111/j.1467-9868.2007.00587.x.

Gneiting, Tilmann and Matthias Katzfuss (2014). "Probabilistic Forecasting". In: *Annual Review of Statistics and Its Application* 1, pp. 125–151. DOI: 10.1146/annurev-statistics-062713-085831.

Gneiting, Tilmann and Adrian E. Raftery (2007). "Strictly Proper Scoring Rules, Prediction, and Estimation". In: *Journal of the American Statistical Association* 102.477, pp. 359–378. DOI: 10.1198/016214506000001437.

Harvey, David, Stephen Leybourne, and Paul Newbold (1997). "Testing the Equality of Prediction Mean Squared Errors". In: *International Journal of Forecasting* 13.2, pp. 281–291. DOI: 10.1016/S0169-2070(96)00719-4.

Hyndman, Rob J and George Athanasopoulos (2021). *Forecasting: Principles and Practice*. 3rd. Melbourne, Australia: OTexts. URL: https://otexts.com/fpp3/.

Hyndman, Rob J. and Yanan Fan (1996). "Sample Quantiles in Statistical Packages". In: *The American Statistician* 50.4, pp. 361–365.

Hyndman, Rob J., Anne B. Koehler, et al. (2008). *Forecasting with Exponential Smoothing: The State Space Approach*. Springer Series in Statistics. Berlin: Springer. ISBN: 978-3-540-71918-2.

Jensen, Vilde, Filippo Maria Bianchi, and Stian Normann Anfinsen (2022). "Ensemble Conformalized Quantile Regression for Probabilistic Time Series Forecasting". Version v2. In: *arXiv preprint arXiv:2202.08756*. Submitted to IEEE Transactions on Neural Networks and Learning Systems. arXiv: 2202.08756 [cs.LG]. URL: https://arxiv.org/abs/2202.08756.

Kingma, Diederik P. and Jimmy Ba (2015). "Adam: A Method for Stochastic Optimization". In: *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*. Originally published as arXiv:1412.6980 [cs.LG].

Lahiri, S. N. (2003). *Resampling Methods for Dependent Data*. Springer Series in Statistics. New York: Springer. ISBN: 978-1-4419-1848-2. DOI: 10.1007/978-1-4757-3803-2.

Petruccelli, Joseph D and Samuel W Woolford (1984). "A consistent test for nonstationarity based on residuals". In: *Journal of the American Statistical Association* 79.387, pp. 611–616.

Politis, Dimitris N. and Joseph P. Romano (1992). "A circular block-resampling procedure for stationary data". In: *Exploring the Limits of Bootstrap*. Ed. by Raoul LePage and Lynne Billard. New York: Wiley, pp. 263–270.

Politis, Dimitris N. and Halbert White (2004). "Automatic Block-Length Selection for the Dependent Bootstrap". In: *Journal of the American Statistical Association* 99.465, pp. 154–164. DOI: 10.1198/016214504000000214.

Salinas, David et al. (2020). "DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks". In: *International Journal of Forecasting* 36.3, pp. 1181–1191. DOI: 10.1016/j.ijforecast.2019.07.001. arXiv: 1704.04110.

Thorarinsdottir, Thordis L. and Nina Schuhen (2017). "Verification: Assessment of Calibration and Accuracy". In: *Norwegian Computing Center* SAMBA/17/17.

Vovk, Vladimir (2019). "Universally consistent conformal predictive distributions". In: *Proceedings of the Eighth Workshop on Conformal and Probabilistic Prediction and Applications (COPA 2019)*. Vol. 105. Proceedings of Machine Learning Research. PMLR, pp. 105–122. URL: http://proceedings.mlr.press/v105/vovk19a.html.

Vovk, Vladimir, Alexander Gammerman, and Glenn Shafer (2005). *Algorithmic Learning in a Random World*. New York: Springer. DOI: 10.1007/b138548.

— (2022). *Algorithmic Learning in a Random World*. Second. Cham, Switzerland: Springer. ISBN: 978-3-031-06648-1. DOI: 10.1007/978-3-031-06649-8.

Vovk, Vladimir, Ilia Nouretdinov, et al. (2017). *Conformal predictive distributions with kernels*. Working paper 20. Working Paper 20. Accessed on January 5, 2026. On-line compression modelling project (new series). URL: http://alrw.net/articles/CPDK.pdf.

West, Kenneth D. (1996). "Asymptotic Inference about Predictive Ability". In: *Econometrica* 64.5, pp. 1067–1084. DOI: 10.2307/2171956.

Ye, Tinghan, Amira Hijazi, and Pascal Van Hentenryck (2025). "Conformal Predictive Distributions for Order Fulfillment Time Forecasting". In: *arXiv preprint arXiv:2505.17340*. Version 2.

Yu, Bin (1994). "Rates of Convergence for Empirical Processes of Stationary Mixing Sequences". In: *The Annals of Probability* 22.1, pp. 94–116.