

UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE CIENCIA
Departamento de Matemática y Ciencia de la Computación



**Estimación de Parámetros para el Modelo GARCH-X a través de un
esquema híbrido de aproximación bayesiana computacional y
aprendizaje profundo**
Tesis de pregrado

Bastián Rafael Díaz Vergara

Profesor guía:

Francisco Plaza Vega

Profesores co-guía:

Andrés Iturriaga

Héctor Araya

Tesis para optar al título de

Ingeniero Estadístico

Santiago - Chile
2025

*Dedicado a todo el que me ha impulsado a seguir aprendiendo y creciendo,
y en especial a mi familia, por su apoyo incondicional.
A mis amigos, por estar siempre ahí.
A mi pareja, por su amor y paciencia.
Y a mis profesores, por compartir su conocimiento y pasión por la ciencia.*

Reconocimientos

Este trabajo se realizó en el marco del Proyecto FONDECYT de Iniciación N.º 11251120 (ANID, Chile). Sus herramientas y/o resultados forman parte de la investigación del proyecto y podrán ser aplicados o extendidos en futuras líneas de estudio vinculadas al mismo.

Agradecimientos

Deseo expresar mi más profundo y sincero agradecimiento a mis padres, cuyo apoyo incondicional y esfuerzo incansable han sido el cimiento sobre el cual he construido cada uno de mis logros. Gracias a su ejemplo, a su entrega y a su confianza, hoy me encuentro culminando esta etapa tan significativa de mi vida. A mi hermano, por la complicidad y el lazo fraterno que siempre me ha sostenido, y por estar ahí en cada momento, escuchándome y acompañándome con paciencia y cariño.

Quiero manifestar mi inmensa gratitud a mi pareja, por caminar a mi lado en los momentos más desafiantes, por su apoyo incondicional y por ser sostén y refugio cuando más lo necesité. También agradezco profundamente a mi madrina, cuyo respaldo y confianza fueron decisivos para que pudiera dar los primeros pasos en mi formación universitaria. Extiendo con afecto

mi reconocimiento a mi mejor amiga, por recordarme que incluso en los momentos de mayor presión es necesario reír, distraerse y respirar. A mis amigos Alberto y Francisca, por las risas y experiencias que hicieron más ligero el camino; a Andrés y Bayron, por ser compañeros ejemplares, con quienes el trabajo y la carrera se volvieron más llevaderos; y a Samantha y Rodolfo, por su generosidad y apoyo en instancias críticas de este proceso. Agradezco con

especial admiración a mis profesores y mentores, quienes han sido guías invaluable en mi desarrollo académico y personal. En particular, al Dr. Francisco Plaza, mi director de tesis, por su paciencia, dedicación y por compartir con generosidad su conocimiento y experiencia en el campo de la estadística y el aprendizaje automático. Su orientación constante y sus valiosas sugerencias han sido fundamentales para la culminación de este trabajo. Finalmente,

agradezco de corazón a todos mis amigos y familiares que, de una u otra forma, estuvieron presentes a lo largo de este recorrido. Su compañía, su ánimo y su cariño fueron faros que iluminaron mi camino y me dieron la fuerza necesaria para alcanzar esta meta.

Resumen

La modelación de procesos con heterocedasticidad condicional exige métodos robustos de estimación de parámetros. En este trabajo se propone un esquema híbrido que integra la Aproximación Bayesiana Computacional (ABC) con redes neuronales profundas para la estimación de parámetros del modelo GARCH-X. Se implementaron y compararon tres variantes principales: (i) ABC-Rejection, (ii) ABC-MCMC clásico con adaptación de escala y calibración de umbrales mediante cuantiles piloto, distancia Mahalanobis. Para ello se diseñaron resúmenes estadísticos basados en una arquitectura neuronal híbrida temporal convolucional y corta-larga memoria, entrenada específicamente para extraer información discriminante de trayectorias simuladas. Los resultados de múltiples réplicas indican que el uso de resúmenes neuronales, junto con esquemas adaptativos de umbral y propuestas más flexibles, mejora significativamente la recuperación de parámetros frente a métodos tradicionales. Además, la incorporación de ajustes de regresión local reduce el sesgo y concentra las distribuciones posteriores alrededor de los valores verdaderos. Este enfoque contribuye a ampliar las herramientas de inferencia bayesiana en contextos donde la verosimilitud es intratable, ofreciendo una vía computacionalmente eficiente y escalable para la estimación en modelos GARCH-X y potencialmente aplicable a otros sistemas estocásticos complejos.

Palabras clave: Aproximación Bayesiana Computacional (ABC); GARCH-X; Redes neuronales profundas; Resúmenes estadísticos; MCMC adaptativo; Estimación de parámetros; Inferencia bayesiana aproximada; Series temporales ambientales.

TABLA DE CONTENIDOS

Resumen	v
1 Formulación del Proyecto	1
1.1 Introducción	1
1.2 Formulación del problema	2
1.3 Objetivos de la investigación	5
1.3.1 Objetivo general	5
1.3.2 Objetivos específicos	5
1.4 Metodología	6
1.5 Resultados Esperados	7
2 Marco Teórico	8
2.1 Series temporales discretas	8
2.1.1 Proceso lineal general	8
2.1.2 Estacionaridad	9
2.1.3 Función de autocovarianza y de autocorrelación parcial	9
2.1.4 Memoria corta y larga	10
2.1.5 Modelos ARMA y ARIMA	10
2.1.6 Análisis de Series de tiempo no estacionarias y heterocedasticidad condicional	11
2.1.7 Modelos GARCH	12
2.1.8 GARCH-X y sus extensiones	12

TABLA DE CONTENIDOS

2.1.9	Estimación por cuasi-máxima verosimilitud (QMV)	14
2.1.10	Representación ARMAX del proceso GARCHX	15
2.2	Aproximación Bayesiana Computacional	16
2.2.1	Algoritmo de Rechazo ABC con estadístico de resumen	17
2.2.2	ABC-MCMC	18
2.2.3	Estadístico de resumen via Redes Neuronales Profundas	20
2.3	Estimación híbrida de parámetros mediante ABC-CNN para modelos GARCH-X	28
3	Estimación híbrida de parámetros para modelo GARCH-X	30
3.1	Redes Neuronales utilizadas como estadístico de Resumen	31
3.1.1	Red Neuronal Convolutacional N°1	31
3.1.2	Red Neuronal CNN + LSTM N°2	33
3.1.3	Red Neuronal Convolutacional N°3	34
3.1.4	TCN + LSTM N°4	36
3.1.5	TCN + LSTM N°5	40
3.2	Comparativa de rendimiento de Estadísticos de Resumen vía DNN para ABC- Rejection	41
3.2.1	Red Neuronal Convolutacional N°1	42
3.2.2	Red Neuronal CNN + LSTM N°2	42
3.2.3	Red Neuronal Convolutacional N°3	43
3.2.4	TCN + LSTM N°4	44
3.2.5	TCN + LSTM N°5	45
3.2.6	Comparación de Resultados	45
3.3	Comparativa de rendimiento de métodos de Aproximación Bayesiana Computacional	46
3.3.1	ABC - Rejection	46
3.3.2	ABC - MCMC	47

TABLA DE CONTENIDOS

4	Aplicación a datos pesqueros del norte de Chile	53
4.1	Contextualización del problema	53
4.2	Análisis exploratorio de datos	56
4.2.1	Descripción de los datos	56
4.3	Visualización de las series temporales	57
4.3.1	Análisis de retornos	59
4.3.2	Test de raíz unitaria	60
4.3.3	Análisis de correlación en los retornos	61
4.3.4	Reducción de dimensionalidad	63
4.3.5	Uniform Manifold Approximation and Projection (UMAP)	65
4.4	Estimación híbrida de parámetros aplicada a los desembarques de Sardina y Anchoqueta	67
4.4.1	Resultados para la Anchoqueta	67
4.4.2	Resultados para la Sardina	69
4.4.3	Comparación entre Anchoqueta y Sardina	70
4.5	Discusión y conclusiones	73
	ANEXOS	76
	Referencias bibliográficas	76
A	Anexo 1: Estimación híbrida de parámetros con Path Signatures	80
A.1	Red Neuronal Convolutiva N°1	80
A.1.1	Descripción de la Arquitectura y Entrenamiento	80
A.1.2	Evolución de Learning Rate	82
A.2	Red Neuronal Convolutiva + LSTM	82
A.2.1	Descripción de la Arquitectura y Entrenamiento	82
A.2.2	Evolución de Learning Rate	85
A.3	Red Neuronal Convolutiva N°3	85
A.3.1	Descripción de la Arquitectura y Entrenamiento	85

TABLA DE CONTENIDOS

A.3.2	Evolución de Learning Rate	87
A.4	Red neuronal TCN + LSTM N°5	87
A.4.1	Descripción de la Arquitectura y Entrenamiento	87
A.4.2	Evolución de Learning Rate	93
B	Anexo 2: Análisis exploratorio de datos	94
B.1	Visualización cuantil cuantil de variables ambientales	94
B.2	Visualización cuantil cuantil de retonos de desembarque de Sardina y Anchoqueta	96
B.3	Codigos del Proyecto	97

ÍNDICE DE FIGURAS

Figura 1.1	Esquema metodológico de la investigación	6
Figura 2.1	Arquitectura típica de una Red Neuronal Convolutacional. Imagen obtenida desde [1].	23
Figura 2.2	Estructura interna de una celda <i>LSTM</i> . Imagen obtenida desde [2]. . .	26
Figura 2.3	Arquitectura típica de una Red Neuronal Convolutacional Temporal. Imagen obtenida desde [3]	27
Figura 3.1	Historia de la pérdida de entrenamiento y validación de la CNN 1D. . .	32
Figura 3.2	Historia de la pérdida de entrenamiento y validación de la CNN + LSTM. 34	
Figura 3.3	Historia de la pérdida de entrenamiento y validación de la CNN 1D con MSE ponderado.	35
Figura 3.4	Historia de la pérdida de entrenamiento y validación de la TCN + LSTM. 41	
Figura 3.5	Resultados del benchmark de la CNN 1D como estadístico de resumen para ABC-Rejection.	42
Figura 3.6	Resultados del benchmark de la CNN + LSTM como estadístico de resumen para ABC-Rejection.	43
Figura 3.7	Resultados del benchmark de la CNN 1D con MSE ponderado como estadístico de resumen para ABC-Rejection.	44
Figura 3.8	Resultados del benchmark de la TCN + LSTM con MSE ponderado como estadístico de resumen para ABC-Rejection.	44
Figura 3.9	Resultados del benchmark de la TCN + LSTM con MSE ponderado como estadístico de resumen para ABC-Rejection.	45

ÍNDICE DE FIGURAS

Figura 3.10 Resultados de una sola réplica de ABC-Rejection	46
Figura 3.11 Cadenas de Markov generadas por el esquema ABC-MCMC para los parámetros del modelo GARCH-X.	50
Figura 3.12 Distribuciones posteriores marginales obtenidas por el esquema ABC- MCMC para los parámetros del modelo GARCH-X.	51
Figura 4.1 Pequeños peces pelágicos.	54
Figura 4.2 Esquema que integra fenómenos locales y globales que influyen en la zona norte de Chile y sus recursos pesqueros, indicando dirección e intensidad de su efecto, obtenido desde (Yáñez, 2008)	55
Figura 4.3 Series temporales de desembarques de sardina y anchoveta, junto con variables ambientales relevantes.	58
Figura 4.4 Series temporales de desembarques de sardina y anchoveta, y sus medias móviles de 12 meses.	59
Figura 4.5 Retornos y retornos logarítmicos de las series de desembarques de Sardina.	59
Figura 4.6 Retornos y retornos logarítmicos de las series de desembarques de Anchoveta.	60
Figura 4.7 ACF y PACF de los retornos y log-retornos de la Sardina	61
Figura 4.8 ACF y PACF de los retornos al cuadrado y log-retornos al cuadrado de la Sardina	62
Figura 4.9 ACF y PACF de los retornos y log-retornos de la Anchoveta	62
Figura 4.10 ACF y PACF de los retornos al cuadrado y log-retornos al cuadrado de la Anchoveta	62
Figura 4.11 Análisis de Componentes Principales (PCA).	64
Figura 4.12 Componente principal dos como serie de tiempo.	64
Figura 4.13 Análisis de Componentes Principales como series de tiempo (PCA).	64
Figura 4.14 Representaciones UMAP de trayectorias transformadas.	66
Figura 4.15 Cadenas de las estimaciones para la Anchoveta.	70

ÍNDICE DE FIGURAS

Figura 4.16 Cadenas de las estimaciones para la Sardina.	71
Figura 4.17 Comparación entre la varianza condicional estimada y los retornos al cuadrado.	72
Figura A.1 Evolución de la tasa de aprendizaje durante el entrenamiento de la CNN 1D.	82
Figura A.2 Evolución de la tasa de aprendizaje durante el entrenamiento de la CNN + LSTM.	85
Figura A.3 Evolución de la tasa de aprendizaje durante el entrenamiento de la CNN 1D N°3.	87
Figura A.4 Evolución de la tasa de aprendizaje durante el entrenamiento de la TCN + LSTM.	93
Figura B.1 Gráfico cuantil-cuantil de las variables ambientales <i>LSAR, LANC, SST</i>	94
Figura B.2 Gráfico cuantil-cuantil de las variables ambientales <i>PDO, SOI, TI</i>	95
Figura B.3 Gráfico cuantil-cuantil de las variables ambientales <i>MEI, N12, N34</i>	95
Figura B.4 Gráfico cuantil-cuantil de la variable <i>HCI</i>	96
Figura B.5 Gráfico cuantil-cuantil de retornos <i>LSAR</i>	96
Figura B.6 Gráfico cuantil-cuantil de retornos <i>LANC</i>	97

ÍNDICE DE TABLAS

Cuadro 3.1	Comparación de R^2 entre parámetros reales y estimados para distintos modelos.	45
Cuadro 3.2	Comparación de R^2 entre parámetros reales y estimados para parámetros de GARCH-X.	52
Cuadro 4.1	Resultados del test ADF para las series de retorno de LSAR	60
Cuadro 4.2	Resultados del test ADF para las series de retorno de LANC	60
Cuadro 4.3	Curtosis de las series de desembarque, retornos y retornos al cuadrado	63

Capítulo 1

Formulación del Proyecto

1.1. Introducción

Una serie temporal es una secuencia de observaciones registradas en instantes específicos de tiempo, considerada como una realización de un proceso estocástico en tiempo discreto.

Estas series se encuentran en múltiples disciplinas (física, medicina, econometría, ciencias ambientales) y permiten estudiar fenómenos como trayectorias, señales fisiológicas, precios financieros, clima y dinámica poblacional.

El análisis de series temporales busca describir, comprender o modelar el mecanismo estocástico subyacente para predecir eventos, proyectar valores futuros e identificar patrones. En el ámbito financiero, muchas series comparten los llamados *hechos estilizados*, como baja autocorrelación de retornos, alta autocorrelación de retornos absolutos, agrupamiento de volatilidad, colas pesadas, efectos de apalancamiento, estacionalidad y heterocedasticidad condicional.

Para modelar la varianza condicional, Engle (1982, [4]) introdujo el modelo *ARCH*, extendido a *GARCH* por Bollerslev (1986, [5]). El *GARCH(1,1)* es ampliamente usado por su capacidad para capturar los principales hechos estilizados. Posteriormente, se incorporaron variables exógenas, originando el modelo y *GARCH-X*, que permite incluir información

relevante adicional y han demostrado utilidad en finanzas y economía.

La estimación de parámetros en *GARCH-X* suele hacerse mediante procedimientos de pseudo-verosimilitud bajo supuestos gaussianos; sin embargo, el no cumplimiento de estos supuestos estrictos, dificulta este enfoque. Como alternativa, se plantea el uso de Aproximación Bayesiana Computacional (ABC), un método que evita calcular la verosimilitud y, en su lugar, aproxima la distribución posterior comparando datos observados con datos simulados a través de estadísticos de resumen. El desafío radica en la falta de estadísticos verdaderamente informativos y en el conocimiento limitado del comportamiento de los parámetros.

Para abordar esto, inspirado en Jiang et al. (2017, [6]), se propone un procedimiento que aprende estadísticos de resumen informativos usando redes neuronales, aprovechando su capacidad de extracción de características y reconocimiento de patrones en series temporales. Este enfoque híbrido *DNN-ABC* busca estimar parámetros cuando se dispone de simulaciones del modelo pero poca información sobre el espacio paramétrico.

Además, se propone una comparación entre los métodos tanto de ABC, explorando diferentes variantes de este método, junto con el estudio de diferentes arquitecturas de redes neuronales profundas como estadístico de resumen para el ABC y posterior estimación de parámetros.

Este trabajo estudia la estimación de parámetros para un modelo *GARCH-X*, analizando sus propiedades, estimación de parámetros, comportamiento en muestras pequeñas mediante simulaciones y aplicaciones con datos reales orientado a la pesquería, como una continuación del trabajo realizado por Plaza y Araya et.al. (2024, [7]).

1.2. Formulación del problema

La estimación **estimación de parámetros** en modelos de series temporales con heterocedasticidad condicional, en contextos donde la verosimilitud es difícil de especificar o su evaluación resulta costosa es un desafío cuando no se logran cumplir supuestos distribucionales necesarios para tal estimación. Los modelos *GARCH-X*, al extender la familia

GARCH con la incorporación de variables exógenas, ofrecen una representación más realista de la dinámica de la varianza condicional en series temporales; sin embargo, este mismo grado de flexibilidad introduce complejidades adicionales en el proceso de estimación, sobre todo bajo innovaciones no gaussianas, colas pesadas o estructuras exógenas poco regulares.

Los métodos tradicionales de máxima verosimilitud o pseudo-verosimilitud, aunque ampliamente utilizados, dependen de supuestos estrictos que rara vez se cumplen. En tales escenarios, la Aproximación Bayesiana Computacional (ABC) emerge como una alternativa, pues evita el cálculo explícito de la verosimilitud y en su lugar aproxima la distribución a posteriori mediante simulaciones del modelo y comparaciones con los datos observados. El reto fundamental de este enfoque es la definición de estadísticas de resumen suficientemente informativas y de baja dimensión, ya que de ellas depende la capacidad de identificar correctamente los parámetros del modelo y garantizar una inferencia confiable.

Para superar esta limitación, se plantea el uso de redes neuronales profundas como herramientas de aprendizaje automático capaces de extraer resúmenes informativos directamente a partir de datos simulados del modelo. Este enfoque híbrido, al que denominamos *DNN-ABC*, combina la flexibilidad de la simulación bayesiana aproximada con la capacidad de las redes neuronales para reconocer patrones y aproximar relaciones complejas en series temporales. La formulación del problema, por tanto, radica en diseñar, implementar y evaluar un procedimiento de inferencia que integre estos componentes, comparando variantes de ABC —como rechazo, *MCMC* y *SMC*—, para determinar su efectividad en escenarios con diferentes grados de complejidad y restricciones de muestra.

En síntesis, este trabajo se centra en el desarrollo y validación de un marco metodológico para la estimación híbrida de parámetros en modelos *GARCH-X*, que busca equilibrar precisión, eficiencia y robustez, abriendo así una vía novedosa para el análisis de series temporales en las que la verosimilitud es parcial, intractable o inadecuada.

Para finalmente aplicar este esquema de estimación híbrida de parámetros al análisis de la variabilidad de los desembarques de Sardina (*Sardinops Sagax*) y Anchoqueta (*Engraulis Ringens*) en la zona norte de Chile, considerando variables exógenas que representen el

CAPÍTULO 1. FORMULACIÓN DEL PROYECTO

impacto de los procesos ambientales, en la pesquería pelágica.

1.3. Objetivos de la investigación

1.3.1. Objetivo general

Desarrollar un esquema híbrido de estimación de parámetros para modelos *GARCH-X*, integrando la Aproximación Bayesiana Computacional (ABC) con redes neuronales profundas utilizadas como generadoras de estadísticas de resumen, con el fin de obtener inferencias robustas en contextos donde la verosimilitud es intratable o poco fiable.

1.3.2. Objetivos específicos

- Implementar un procedimiento de estimación híbrido **ABC–DNN** aplicado a modelos *GARCH-X*, siguiendo los lineamientos metodológicos descritos en la literatura reciente.
- Comparar distintas arquitecturas de redes neuronales profundas (*CNN*, *LSTM*, *TCN*) para la construcción de estadísticas de resumen informativas en el marco del ABC.
- Evaluar el desempeño de diferentes variantes de ABC en la estimación de parámetros de modelos *GARCH-X*, analizando su eficiencia, precisión y robustez.
- Analizar registros históricos de los desembarques de Sardina y Anchoqueta, además de las variables ambientales a considerar en el análisis a partir de repositorios nacionales e internacionales.
- Aplicar el esquema de estimación seleccionado para el modelo *GARCH-X* utilizando los datos recopilados de pesquerías pelágicas del norte de Chile y su interacción con el ambiente para explicar la variabilidad de los desembarques de Anchoqueta (*Engrauis Ringens*) y Sardina (*Sardinops Sagax*).

1.4. Metodología

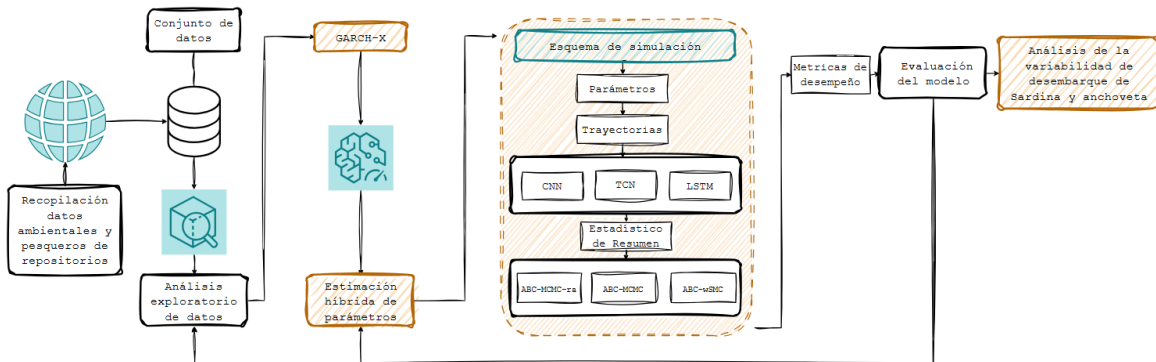


Figura 1.1: Esquema metodológico de la investigación

El esquema metodológico descrito propone un enfoque integrado para analizar la variabilidad en los desembarques de Sardina y Anchoveta, considerando factores ambientales y pesqueros, tales como (SERNAPESCA, 2024 [8] ; DIRECTEMAR, 2024 [9] ; NOAA, 2024 [10]). El proceso comienza con la recopilación de datos provenientes de repositorios especializados, los cuales incluyen información tanto ambiental como pesquera. Estos datos son sometidos a un análisis exploratorio inicial, cuyo objetivo es identificar patrones, tendencias y posibles inconsistencias, constituyendo una base sólida para la construcción de modelos.

Posteriormente, los datos procesados se almacenan en una base estructurada y son utilizados en la aplicación de un modelo *GARCH-X*, diseñado para analizar la volatilidad en función de variables exógenas relevantes. Este modelo permite realizar una estimación híbrida de parámetros, método originalmente propuesto en (Plaza, 2024) [7], el cual se busca generalizar, combinando arquitecturas y comparando diferentes *ABC*, combinando así métodos estadísticos y computacionales para mejorar la precisión del ajuste.

La estimación híbrida de parámetros comprende de varias etapas, en primer lugar, la etapa de simulación, que incluye la generación de trayectorias del modelo *GARCH-X* que

representan las dinámicas de las series temporales.

Estas trayectorias son la entrada a un modelo de Deep Learning, que actúa como un estadístico de resumen, extrayendo características relevantes de las series temporales simuladas. Este modelo de Deep Learning se entrena para aprender patrones complejos y con ello se pretende capturar estadísticos de resumen, para que con el algoritmo de Aproximación Bayesiana Computacional (ABC) facilitando la estimación de parámetros y su inferencia.

El desempeño del modelo es evaluado utilizando métricas específicas que aseguran su robustez y precisión. Finalmente, el modelo validado se compara con de la variabilidad en los desembarques de Sardina (*Sardinops Sagax*) y Anchoveta (*Engraulis Ringens*), proporcionando información detallada sobre los factores que influyen en sus dinámicas poblacionales y permitiendo una mejor comprensión de su relación con el ambiente marino. Este enfoque metodológico combina técnicas tradicionales y herramientas avanzadas de modelado y simulación, ofreciendo una perspectiva integral para abordar problemas complejos en el ámbito de la pesquería pelágica.

1.5. Resultados Esperados

Se espera que la incorporación y comparación de distintas arquitecturas de redes neuronales profundas, integradas con diversos métodos de Aproximación Bayesiana Computacional (ABC), permita optimizar el proceso híbrido de estimación de parámetros en modelos *GARCH-X*. Este enfoque busca lograr una representación precisa de la volatilidad de los desembarques de Sardina (*Sardinops sagax*) y Anchoveta (*Engraulis ringens*), facilitando la comprensión de cómo los factores ambientales y pesqueros influyen en sus dinámicas poblacionales en el norte de Chile, y aportando herramientas para la explotación sostenible de estos recursos. Asimismo, la investigación validará un marco metodológico para la estimación de parámetros en modelos de series de tiempo, con potencial de aplicación en otros contextos y disciplinas.

Capítulo 2

Marco Teórico

El estudio de la variabilidad espacio temporal en sistemas complejos exige herramientas que capten de manera rigurosa la dependencia temporal de las observaciones. Con ello, en la búsqueda de métodos de estimación más flexibles, se ha desarrollado una amplia gama de modelos, que permiten capturar la dinámica de las series temporales complejas. Entonces en este capítulo se exploran los fundamentos que sustentan una estimación híbrida de parámetros, para un modelo particular de series temporales discretas.

2.1. Series temporales discretas

Una serie temporal discreta es un conjunto de observaciones X_t $\{t \in \mathbb{Z}\}$, de observaciones registradas en instantes equiespaciados. En este contexto, cada observación puede interpretarse como la realización de un proceso estocástico definido en tiempo discreto.

2.1.1. Proceso lineal general

Un proceso lineal general es definido en (Palma, 2016 [11]) como un conjunto finito de observaciones X_t que puede ser escrito como

$$X_t = \psi(B)\varepsilon_t = \sum_{j=-\infty}^{\infty} \psi_j \varepsilon_{t-j}, \quad t \in \mathbb{Z}, \quad (2.1)$$

donde B es el operador de rezago ($Y_t = Y_{t-1}$) y ε_t es un ruido blanco con media cero y varianza constante. La condición $\sum_{j=-\infty}^{\infty} \psi_j^2 < \infty$ garantiza la convergencia de la serie y la existencia de los momentos de segundo orden.

2.1.2. Estacionariedad

Sea $X_t \{t \in \mathbb{Z}\}$ un proceso definido sobre un mismo espacio de probabilidad, distinguiéndose dos nociones principales de estacionariedad (Brockwell, 2009[12]):

2.1.2.1. Estacionariedad de primer orden

El proceso es *estrictamente estacionario* si, para todo $k \in \mathbb{N}$ y $h \in \mathbb{Z}$, los vectores $(X_1, \dots, X_k)^\top$ y $(X_{1+h}, \dots, X_{k+h})^\top$ comparten la misma distribución conjunta.

2.1.2.2. Estacionariedad de segundo orden

El proceso X_t se dice estacionario de segundo orden o estacionario débil si cumple,

(i) $\mathbb{E}(X_t^2) < \infty, \quad \forall t \in \mathbb{Z}$

(ii) $\mathbb{E}(X_t) = m, \quad \forall t \in \mathbb{Z}$

(iii) $\mathbb{Cov}(X_r, X_s) = \mathbb{Cov}(X_{r+k}, X_{s+k}) \quad \forall r, s, k \in \mathbb{Z}$

Así, un proceso estacionario de primer orden implica la de segundo orden; sin embargo, un proceso estacionario de segundo orden no es de primer orden necesariamente.

2.1.3. Función de autocovarianza y de autocorrelación parcial

Para procesos estacionarios de segundo orden, la *función de autocovarianza*

$$\gamma(h) = \text{Cov}(X_t, X_{t+h}), \quad h \in \mathbb{Z}, \quad (2.2)$$

describe la dependencia lineal entre observaciones separadas h periodos. De ella, desprende la *función de autocorrelación (ACF)*, se define como

$$\rho(h) = \gamma(h)/\gamma(0).$$

La *función de autocorrelación parcial (PACF)*, denotada $\alpha(h)$, mide la correlación entre X_t y X_{t+h} una vez eliminada la influencia de los rezagos intermedios (Box, 2015 [13]). Se obtiene comúnmente como el coeficiente h -ésimo de un modelo autorregresivo ajustado de orden h .

2.1.4. Memoria corta y larga

Un proceso presenta *memoria corta* si $\sum_{h=0}^{\infty} |\rho(h)| < \infty$; de lo contrario, se dice que posee *memoria larga*, caracterizada por autocorrelaciones que decrecen según una ley potencial (Palma, 2007 [14]).

2.1.5. Modelos ARMA y ARIMA

Un proceso $ARMA(p, q)$ combina una parte autorregresiva (AR) y otra de medias móviles (MA):

$$X_t = \omega + \sum_{i=1}^p \phi_i X_{t-i} + \varepsilon_t + \sum_{j=1}^q \theta_j \varepsilon_{t-j}, \quad \varepsilon_t \sim \text{WN}(0, \sigma^2). \quad (2.3)$$

Si la serie requiere d diferenciaciones para alcanzar estacionaridad, el modelo resultante es un $ARIMA(p, d, q)$, que se define como un proceso:

$$\Phi(B)(1 - B)^d X_t = \Theta(B)\varepsilon_t, \quad (2.4)$$

con Φ y Θ polinomios de orden p y q , respectivamente (Shumway, 2006 [15]).

2.1.6. Análisis de Series de tiempo no estacionarias y heterocedasticidad condicional

Existe un interés en modelar series cronológicas que cumplen ciertas *propiedades estadísticas* observadas empíricamente cuyo origen proviene de series de tiempo financieras, tales propiedades fueron denominadas *hechos estilizados* que comparten las series financieras (Franco y Zakoian, 2019 [16]).

Sea P_t el precio de cierre de un activo en el instante $t \in \mathbb{Z}$; definimos el *log-retorno* como

$$r_t = \log\left(\frac{P_t}{P_{t-1}}\right)$$

de modo que $r_t = \log(1 + R_t)$ con $R_t = \frac{P_t - P_{t-1}}{P_{t-1}}$ el retorno porcentual.

2.1.6.0.1. Hechos estilizados *Los hechos estilizados* han sido estudiados y documentados en series financieras demostrado las siguientes características:

- *Series no estacionarias*: Generalmente las series son descritas por una caminata aleatoria sin intercepto, sin embargo las trayectorias de los retornos de los precios son generalmente compatibles con la estacionariedad de segundo orden.
- *Ausencia de autocorrelación en la variación de retornos*: Las series de variaciones de precio r_t definidas previamente, generalmente muestran autocorrelaciones bajas o nulas, haciéndolas similares a procesos de ruidos blanco.
- *Autocorrelaciones cuadradas en la variación de r_t* : Los retornos al cuadrado (r_t^2) o los retornos absolutos ($|r_t|$) presentan una autocorrelación significativa. Esta propiedad no es incompatible con la definición de ruido blanco sin embargo, muestra que tal propiedad no es fuerte.
- *Clusters de volatilidad*: Grandes retornos absolutos tienden a aparecer en *clusters*. Generalmente, se detectan de manera visual y a pesar de su ocurrencia, no se puede establecer periodicidad respecto a ellos.

- *Distribuciones de colas pesadas*: La distribución empírica de retornos diarios no parece tener una distribución Gaussiana. Suelen tener densidades de forma leptocúrtica de colas pesadas.
- *Efectos de apalancamiento*: Corresponde a una asimetría en el impacto de los retornos negativos respecto a los positivos, donde los negativos, impactan más en la volatilidad los retornos positivos de la misma magnitud.
- *Estacionalidad*: Característica que corresponde a series regidas y medidas por efectos del calendario.

Estas características violan la hipótesis de varianza condicional constante, dando lugar a los modelos *ARCH* y *GARCH*.

2.1.7. Modelos GARCH

El modelo *GARCH*(p, q) se define como

$$X_t = \sigma_t \eta_t, \quad \sigma_t^2 = \omega + \sum_{i=1}^q \alpha_i r_{t-i}^2 + \sum_{j=1}^p \beta_j \sigma_{t-j}^2, \quad (2.5)$$

con η_t *i.i.d.* de media cero y varianza unitaria. Requiere $\omega > 0$ y coeficientes no negativos que satisfagan $\sum \alpha_i + \sum \beta_j < 1$ para asegurar estacionaridad condicional.

2.1.8. GARCH-X y sus extensiones

Los modelos *GARCH-X* enriquecen la especificación tradicional de (2.5) incorporando una o más covariables externas que contribuyen a explicar la dinámica de la varianza condicional σ_t^2 . Una formulación general es de un modelo *GARCH-X* de orden p, q, s es:

$$X_t = \sigma_t \eta_t, \quad (2.6)$$

$$\sigma_t^2 = \omega + \sum_{i=1}^q \alpha_i X_{t-i}^2 + \sum_{j=1}^p \beta_j \sigma_{t-j}^2 + \sum_{k=1}^s \delta_k Z_{t-k}^2 \quad (2.7)$$

donde Z_t es la covariable, δ su parámetro asociado que pondera el efecto de la variable exógena.

2.1.8.1. Modelo LW-GARCH

Tudor [17] propone el modelo *Liquidity-Weighted GARCH*, en el que el número de transacciones L_t pondera la volatilidad:

$$X_t = \sigma_t \eta_t, \quad \sigma_t^2 = \omega + \sum_{i=1}^q \alpha_i r_{t-i}^2 + \sum_{j=1}^p \beta_j \sigma_{t-j}^2 + \sum_{k=1}^s \delta_k L_{t-k}^2 \quad (2.8)$$

siendo $\{\eta_t\}$ *i.i.d.* con media cero y varianza unitaria.

2.1.8.2. Modelo LW-EGARCH

Una versión logarítmica asimétrica del modelo anterior 2.8 extiende a EGARCH:

$$\log \sigma_t^2 = \alpha_0 + \sum_{i=1}^q \alpha_i g(\eta_{t-i}) + \sum_{j=1}^p \beta_j \log \sigma_{t-j}^2 + \sum_{k=1}^s \gamma_k \log L_{t-k}, \quad (2.9)$$

donde $g(\eta_{t-i}) = \theta \eta_{t-i} + \gamma (|\eta_{t-i}| - \mathbb{E}|\eta_{t-i}|)$ captura asimetrías de apalancamiento (Tudor et al., 2014 [18]).

2.1.8.3. Modelo GARCH-FunXL

Fuest and Mittnik [19] introduce una covariable funcional $y_{t,i}(m)$ que describe la liquidez intradía en el punto $m \in \mathcal{P} \subset \mathbb{R}$:

$$\sigma_{t,i}^2 = \omega + \alpha_1 r_{t,i-1}^2 + \beta_1 \sigma_{t,i-1}^2 + \int_{\mathcal{P}} \gamma(m) y_{t,i-1}(m) dm, \quad i = 1, \dots, T_{\text{día}}.$$

2.1.8.4. Modelo log-GARCH-X

Otra formulación ampliamente usada es el log-GARCH con efectos exógenos:

$$\log \sigma_t^2 = \omega + \sum_{i=1}^p \alpha_i \log r_{t-i}^2 + \sum_{j=1}^q \beta_j \log \sigma_{t-j}^2 + g(\lambda, Z_t),$$

donde Z_t representa la covariable y $g(\cdot, \cdot)$ una función ligada al parámetro λ (Sucarrat et. al. , 2016 [20]).

2.1.9. Estimación por cuasi-máxima verosimilitud (QMV)

Los parámetros de (2.5) y (2.6) se estiman típicamente mediante el método de *cuasi-máxima verosimilitud* (QM), que es consistente y asintóticamente normal incluso bajo desviaciones de la normalidad [16]. Dados los valores iniciales apropiados, el estimador $\hat{\theta}_n$ maximiza la log-verosimilitud condicional

$$\ell_n(\theta) = -\frac{1}{2} \sum_{t=1}^n [\log(2\pi\tilde{\sigma}_t^2) + r_t^2/\tilde{\sigma}_t^2]. \quad (2.10)$$

Para que el estimador de cuasi-máxima verosimilitud (QML) sea consistente y asintóticamente normal en un modelo $GARCH(p, q, s)$ -X, se requiere que:

- El vector verdadero de parámetros θ_0 pertenezca al interior del espacio Θ y el modelo esté correctamente especificado en su media $\mu_t(\theta)$ y varianza condicional $\sigma_t^2(\theta)$.

- Las series $\{X_t\}$, y de existir, $\{Z_t\}$ sean estrictamente estacionarias, ergódicas y posean momentos finitos (orden 2 para consistencia, orden 4 para normalidad asintótica), cumpliendo además $\sum_{i=1}^q \alpha_i + \sum_{j=1}^p \beta_j < 1$.
- Las innovaciones $\{\eta_t\}$ sean *i.i.d.*, centradas $\mathbb{E}[\eta_t] = 0$, de varianza unitaria $\mathbb{E}[\eta_t^2] = 1$ y con densidad continua, y que Z_t sea \mathcal{F}_{t-1} -medible o independiente de η_t .
- La log-verosimilitud condicional $\ell_t(\theta)$ sea dos veces diferenciable en θ y su matriz de información

$$I(\theta_0) = \mathbb{E} \left[-\frac{\partial^2 \ell_t(\theta_0)}{\partial \theta \partial \theta^\top} \right]$$

sea finita y definida positiva.

- El modelo sea identificable, es decir, los polinomios autorregresivo y de medias móviles no compartan raíces comunes.

Bajo estos supuestos resulta [16]

$$\widehat{\theta}_n \xrightarrow{p} \theta_0, \quad \sqrt{n}(\widehat{\theta}_n - \theta_0) \xrightarrow{d} \mathcal{N}(0, I(\theta_0)^{-1}),$$

2.1.10. Representación ARMAX del proceso GARCHX

Existe una directa correspondencia entre los procesos del tipo *GARCH* y *ARMA*, ya que un proceso del tipo *GARCHX* al cuadrado es representado como un proceso *ARMA*, donde para sus procesos homólogos con variable exógena, esta relación se mantiene, En esta representación Ilmonen et al. [21], el cuadrado de la serie X_t se expresa como una combinación lineal de su rezago, un término de ruido estructurado, el rezago de este último y una variable exógena al cuadrado,

$$X_t^2 = \omega + \phi X_{t-1}^2 + \eta_t + \theta \eta_{t-1} + \delta Z_{t-1}^2,$$

donde $\eta_t = (\varepsilon_t^2 - 1)\sigma_t^2$, $\phi = \alpha + \beta$, $\theta = -\beta$, y se cumple que $\mathbb{E}(\eta_t) = 0$.

A pesar de que esta formulación tiene la apariencia de un modelo ARMAX convencional, existe una diferencia clave respecto a lo planteado en [21], en este caso, la esperanza del término exógeno al cuadrado es distinto de uno, es decir,

$$\mathbb{E}(Z_t^2) = C_0 \quad \text{con} \quad C_0 \neq 1.$$

Esto implica que los parámetros del modelo no pueden ser estimados aplicando directamente métodos estándar de modelos autorregresivos con variables exógenas, lo que justifica la necesidad de un nuevo enfoque de estimación.

2.2. Aproximación Bayesiana Computacional

Dentro del marco de la Inferencia Bayesiana, se enmarca la Aproximación Bayesiana Computacional, la cual es una técnica que permite hacer inferencia aproximada inclusive cuando es no es posible derivar una función de verosimilitud expícita (Sisson y Fan, 2018[22]). El método *ABC*, utiliza simulaciones desde la distribución *a priori* para aproximar la distribución a posteriori, comparando los datos observados con los datos simulados. para datos discretos el algoritmo de ABC por rechazo se define como, El Algoritmo ?? [6]no puede

Algoritmo 1 ABC rejection sampling

- 1: **para** $i = 1, \dots, n$ **hacer**
 - 2: **repetir**
 - 3: Proponer $\theta' \sim \pi(\theta)$
 - 4: Elegir $X' \sim \mathcal{M}$ dado θ'
 - 5: **hasta que** $X' = x_{\text{obs}}$ (*Criterio de Aceptación*)
 - 6: Accept θ' and let $\theta^{(i)} = \theta'$
 - 7: **fin para**
-

aceptar muestras cuando X' y x_{obs} son continuos, ya que $X' = x_{\text{obs}}$ ocurre con probabilidad cero. Para solucionarlo, se permite una tolerancia ε mediante $\|A' - x_{\text{obs}}\| < \varepsilon$.

2.2.1. Algoritmo de Rechazo ABC con estadístico de resumen

En problemas donde los datos observados y_{obs} son de alta dimensión, el procedimiento clásico de ABC por rechazo presenta dos dificultades fundamentales: la tasa de aceptación de simulaciones se desploma y el coste computacional crece en exceso. Un enfoque útil para abordar este problema (6; 22), reemplaza la comparación directa de vectores de observaciones por la de un estadístico de resumen

$S: \mathcal{Y} \rightarrow \mathbb{R}^d$ de baja dimensión $d \ll \dim(y)$, y se define un umbral de tolerancia $\varepsilon > 0$. Así, en lugar de requerir $\|Y' - y_{\text{obs}}\| \leq \varepsilon$, se acepta una simulación $Y' \sim p(\cdot | \theta)$ si $\|S(Y') - S(y_{\text{obs}})\| \leq \varepsilon$.

De este modo, la densidad posterior aproximada queda:

$$\pi_{\text{ABC}}(\theta | y_{\text{obs}}) \propto \int \mathbb{I}(\|S(y) - S(y_{\text{obs}})\| \leq \varepsilon) p(y | \theta) \pi(\theta) dy$$

donde $\mathbb{I}(\cdot)$ es la función indicadora.

Con ello, se debe encontrar un equilibrio entre el umbral de aceptación ε . Un valor más pequeño acerca la aproximación π_{ABC} a la verdadera posterior $\pi(\theta | y_{\text{obs}})$, pero reduce la tasa de aceptación y eleva la varianza. Por el contrario, un ε más alto mejora la eficiencia de muestreo a costa de introducir sesgo en la inferencia.

De la misma forma elegir un Estadístico de resumen S de dimensión muy baja acelera el criterio de aceptación pero puede descartar información crucial, incrementando el error de aproximación. En cambio, resúmenes más informativos capturan mejor la estructura de los datos, pero añaden coste computacional.

En el límite $\varepsilon \rightarrow 0$, se demuestra que

$$\lim_{\varepsilon \rightarrow 0} \pi_{\text{ABC}}(\theta | y_{\text{obs}}) = \pi(\theta | y_{\text{obs}})$$

, implicando que la aproximación se vuelve exacta a medida que ε se acerca a cero. De esta

forma, el algoritmo de ABC por rechazo con estadístico de resumen se puede expresar como:

Algoritmo 2 ABC por rechazo

```
1: para  $i = 1$  hasta  $N$  hacer
2:   Muestra  $\theta' \sim \pi(\theta)$ 
3:   Simula  $Y' \sim p(y \mid \theta')$ 
4:   si  $|S(Y') - S(y_{\text{obs}})| \leq \varepsilon$  entonces
5:     Acepta  $\theta'$  y asigna  $\theta^{(i)} = \theta'$ 
6:   si no
7:     Rechaza  $\theta'$  y repite
8:   fin si
9: fin para
```

2.2.2. ABC-MCMC

El algoritmo *Markov chain Monte Carlo* sin verosimilitud (*ABC-MCMC*) traslada la idea del algoritmo *Metropolis-Hastings* a un contexto donde la verosimilitud $p(y_{\text{obs}} \mid \theta)$ es intratable. Partiendo de un valor inicial $\theta^{(0)}$, cada iteración t realiza¹:

¹Algoritmo sintetizado a partir de [23].

Algoritmo 3 ABC-MCMC

Entrada: Observación y_{obs} , prior $\pi(\theta)$, propuesta $q(\theta' | \theta)$, distancia ρ , tolerancia ε , número de iteraciones T

Salida: Muestras $\{\theta^{(t)}\}_{t=1}^T$ que siguen aproximadamente $\pi_{\text{ABC}}(\theta | y_{\text{obs}})$

- 1: Inicializar $\theta^{(0)} \sim \pi(\theta)$
 - 2: **para** $t = 0$ **hasta** $T - 1$ **hacer**
 - 3: Proponer $\theta' \sim q(\cdot | \theta^{(t)})$
 - 4: Simular $Y' \sim p(\cdot | \theta')$
 - 5: Calcular $\rho = \rho(S(Y'), S(y_{\text{obs}}))$
 - 6: **si** $\rho \leq \varepsilon$ **entonces**
 - 7: $\alpha \leftarrow \min\left\{1, \frac{\pi(\theta') q(\theta^{(t)} | \theta')}{\pi(\theta^{(t)}) q(\theta' | \theta^{(t)})}\right\}$
 - 8: Generar $u \sim \mathcal{U}(0, 1)$
 - 9: **si** $u \leq \alpha$ **entonces**
 - 10: $\theta^{(t+1)} \leftarrow \theta'$
 - 11: **si no**
 - 12: $\theta^{(t+1)} \leftarrow \theta^{(t)}$
 - 13: **fin si**
 - 14: **si no**
 - 15: $\theta^{(t+1)} \leftarrow \theta^{(t)}$
 - 16: **fin si**
 - 17: **fin para**
-

La cadena estacionaria genera muestras aproximadas de $\pi_{\text{ABC}}(\theta | y_{\text{obs}})$; sin embargo, el método puede mezclar lento en las colas de la posterior, pues la probabilidad de simular Y' cercano a y_{obs} decrece rápidamente cuando θ' está lejos de las regiones de alta densidad.

2.2.3. Estadístico de resumen via Redes Neuronales Profundas

Cuando se incrementa la dimensión de las estadísticas resumen, la precisión de los métodos ABC disminuye. Se ha demostrado que el error de estimación (Sisson, 2018 [22]) decrece más lentamente a medida que aumenta la dimensión de los datos simulados, incluso bajo condiciones óptimas. Por ello, al usar estadísticas resumen de muy alta dimensión generalmente produce resultados no favorables.

Un estadístico de resumen *suficiente* es una opción ya que implica que su error de aproximación es cero, (Kolmogorov, 1942[24]). Con ello, un estadístico, es suficiente si contiene toda la información sobre los parámetros del modelo. En el contexto bayesiano, esto significa que la distribución posterior condicional de los parámetros dada la estadística es igual a la posterior dada los datos completos. Sin embargo, el teorema de Pitman–Koopman–Darmois (Barankin , 1963[25]) indica que solo en modelos de familia exponencial existen estadísticas suficientes de baja dimensión . Por lo tanto, para la mayoría de los modelos, no existen tales estadísticas, y se deben usar resúmenes insuficientes pero informativos.

Un enfoque para capturar un estadístico de resumen suficiente es via redes neuronales profundas (Jiang , 2017 [6]), automatizando el proceso de construcción del estadístico, al entrenar un modelo de red neuronal, prediciendo así parámetros de datos sintéticos. De esta forma el estadístico de resumen corresponde a las medias a posteriori de los parámetros. El enfoque propone, entrenar una red neuronal profunda (*DNN*), para capturar un estadístico de resumen para un Y de alta dimensión, donde se espera que esta red, sea una aproximación efectiva de las media a posteriori $\mathbb{E}_\pi[\theta | Y]$, donde al construir un estimador con error cuadrático medio mínimo $\hat{\theta}(Y)$ en un set de datos generado por el modelo, $\{(\theta^{(i)}, Y^{(i)})\}_{1 \leq i \leq N} \sim \pi \times \mathcal{M}$. De esta forma el problema de viene dado por la expresión,

$$\min_{\beta} \frac{1}{N} \sum_{i=1}^N \| f_{\beta}(Y^{(i)} - \theta^{(i)}) \|^2$$

,

donde f_β denota la red neuronal profunda y el estimador $\hat{\theta}(Y) = f_\beta(Y)$ se aproxima a $\mathbb{E}_\pi[\theta | Y]$ por lo tanto es útil como estadístico de resumen para el ABC correspondiente.

2.2.3.1. Arquitecturas de Redes Neuronales Profundas

Existen diferentes tipos de redes neuronales profundas que pueden ser utilizadas como estadísticos de resumen, entre ellas se encuentran las redes neuronales convolucionales (*CNN*) y las redes neuronales recurrentes (*RNN*), entre otras.

2.2.3.1.1. Redes Neuronales Convolucionales (*CNN*) Las Redes Neuronales Convolucionales (*CNNs* o *ConvNets*) son una clase de redes neuronales diseñadas específicamente para procesar datos con una topología de cuadrícula, como una imagen 2D (Goodfellow et al., 2016 [26]). A diferencia de las redes que aplanan la entrada en un vector, las *CNNs* aprovechan la estructura espacial de los datos mediante operaciones matemáticas especializadas para construir jerarquías de características.

El componente central de una *CNN* es la *capa convolucional*, que utiliza la operación de convolución para detectar características locales. Esta operación se realiza deslizando un filtro o kernel, que es una pequeña matriz de pesos, a través de toda la información.

Matemáticamente, si la entrada es información I y el kernel es K , la convolución discreta que produce un mapa de características S en la posición (i, j) se define como:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i - m, j - n)K(m, n)$$

Esta operación se basa en tres principios clave:

- iteraciones dispersas: Cada unidad del mapa de características de salida solo se conecta a una pequeña región de la entrada (el campo receptivo), reduciendo el número de parámetros.
- Compartición de parámetros: El mismo kernel K se reutiliza en toda la información disponible, permitiendo que la red detecte el mismo patrón sin importar su ubicación

espacial.

- Representaciones equivariantes a la traslación: Si la entrada se traslada, la salida se traslada de la misma manera, lo que ayuda a la robustez del modelo.

Una CNN moderna se compone de una secuencia de capas, cada una con una función matemática específica. Inmediatamente después de cada operación de convolución, se aplica una función de activación no lineal. Sin esta no linealidad, La función de activación más utilizada es la *Unidad Lineal Rectificada (ReLU)*. La función ReLU se define como:

$$f(x) = \max(0, x)$$

Esta función introduce no linealidad de una manera computacionalmente eficiente: devuelve 0 si la entrada x es negativa y devuelve x sin cambios si es positiva. La capa de agrupación o pooling tiene como objetivo reducir las dimensiones espaciales (ancho y alto) del mapa de características. Esto reduce la carga computacional y el riesgo de sobreajuste, además de proporcionar un grado de invarianza a pequeñas traslaciones. La operación más común es el *Max Pooling*. Esta operación divide el mapa de características en una cuadrícula de regiones no superpuestas y, para cada región, produce como salida el valor máximo. Para una región de entrada P , la operación se define como:

$$output = \max_{i,j \in P} (x_{ij})$$

Al seleccionar solo las activaciones más fuertes, el max pooling ayuda a retener las características más prominentes detectadas por la capa de convolución.

Al final de la arquitectura, después de varias capas convolucionales y de pooling, los mapas de características resultantes se aplanan en un vector unidimensional. Este vector se introduce en una o más capas completamente conectadas (como las de una red neuronal tradicional) para realizar la clasificación o regresión final.

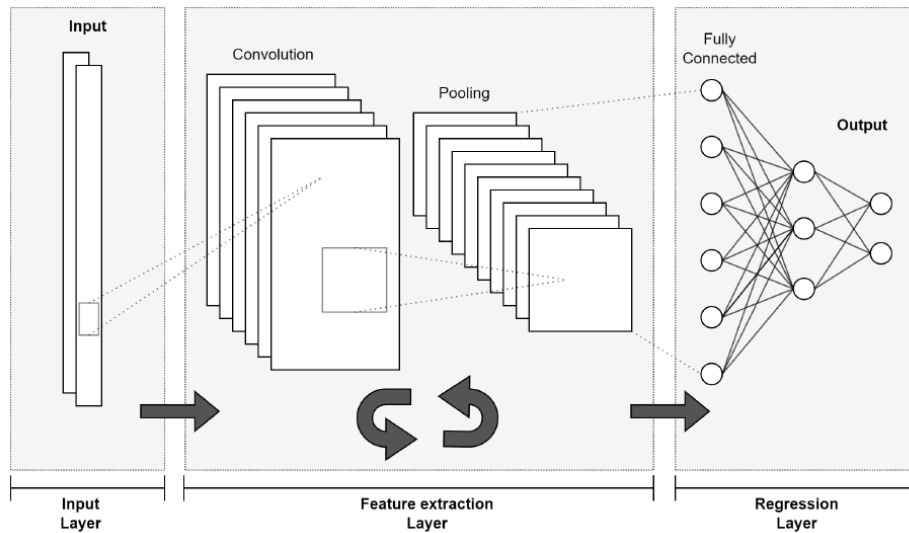


Figura 2.1: Arquitectura típica de una Red Neuronal Convolutiva. Imagen obtenida desde [1].

2.2.3.1.2. Redes de Memoria a Corto y Largo Plazo (LSTM) Las Redes Neuronales Recurrentes (RNN) estándar están diseñadas para procesar secuencias de datos, pero sufren de un problema fundamental conocido como el gradiente desvaneciente o explosivo (*vanishing/exploding gradients*). Este problema limita su capacidad para aprender y recordar dependencias a largo plazo en una secuencia.

Para abordar esta limitación, Hochreiter y Schmidhuber (1997) [27] introdujeron las Redes de Memoria a Corto y Largo Plazo (LSTM). Las LSTM son un tipo avanzado de RNN que utilizan un mecanismo de compuertas (*gating mechanism*) para regular el flujo de información, permitiendo que la red aprenda qué información almacenar en la memoria, qué información olvidar y qué información usar para la salida en cada paso de tiempo.

El componente central de una red LSTM es la celda de memoria, que mantiene un estado de celda (C_t) a lo largo del tiempo. Este estado actúa como una cinta transportadora de información, con la capacidad de ser modificado por tres compuertas principales: la de olvido, la de entrada y la de salida.

Para un paso de tiempo t , la celda LSTM toma como entrada la entrada actual X_t y el estado oculto anterior H_{t-1} .

La primera compuerta decide qué información del estado de celda anterior (C_{t-1}) debe ser descartada. Esta decisión se basa en la entrada actual (X_t) y el estado oculto anterior (H_{t-1}).

La salida de la compuerta de olvido, F_t , es un vector con valores entre 0 y 1. Un valor de 1 significa "mantener esta información", mientras que un valor de 0 significa "olvidar esta información".

Matemáticamente, se define como:

$$F_t = \sigma(X_t W_{xf} + H_{t-1} W_{hf} + b_f)$$

donde:

- σ es la función de activación sigmoide, que mapea los valores al rango $[0, 1]$.
- W_{xf} y W_{hf} son las matrices de pesos para la entrada y el estado oculto, respectivamente.
- b_f es el vector de sesgo (bias).

La compuerta de entrada determina qué nueva información se almacenará en el estado de celda. Este proceso tiene dos partes:

1. Decidir qué valores actualizar: La compuerta de entrada, I_t , utiliza una función sigmoide para decidir qué partes de la nueva información son importantes.

$$I_t = \sigma(X_t W_{xi} + H_{t-1} W_{hi} + b_i)$$

que podrían ser añadidos al estado de la celda.

$$\tilde{C}_t = \tanh(X_t W_{xc} + H_{t-1} W_{hc} + b_c)$$

El estado de celda anterior, C_{t-1} , se actualiza al nuevo estado C_t combinando los resultados

de las compuertas de olvido y de entrada.

$$C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t,$$

donde:

- $F_t \odot C_{t-1}$ olvida la información que la compuerta de olvido decidió descartar.
- $I_t \odot \tilde{C}_t$ agrega la nueva información candidata, escalada por la importancia determinada por la compuerta de entrada.

Finalmente, la compuerta de salida decide cuál será la salida de la celda (el nuevo estado oculto H_t). La salida se basa en el estado de celda actualizado (C_t), pero filtrado por la compuerta de salida. En primer lugar, se decide qué partes del estado de la celda se van a emitir como salida:

$$O_t = \sigma(X_t W_{xo} + H_{t-1} W_{ho} + b_o)$$

y se multiplica por la salida de la compuerta de salida O_t :

$$H_t = O_t \odot \tanh(C_t)$$

Este H_t es la salida de la celda LSTM en el paso de tiempo t , también se utiliza como entrada para el siguiente paso de tiempo ($t + 1$). Gracias a este mecanismo, las LSTM pueden controlar de manera efectiva el flujo de gradientes y capturar dependencias a muy largo plazo en los datos secuenciales (Zhang, 2021 [2]).

2.2.3.1.3. Redes Neuronales Convolucionales Temporales Las *Redes Neuronales Convolucionales Temporales* (TCN, *Temporal Convolutional Networks*) de las redes convolucionales con el respeto estricto a la causalidad temporal. A diferencia de las redes recurrentes tradicionales, las *TCN* no dependen de la propagación de estados ocultos a lo largo del tiempo sino que modelan las dependencias temporales mediante convoluciones

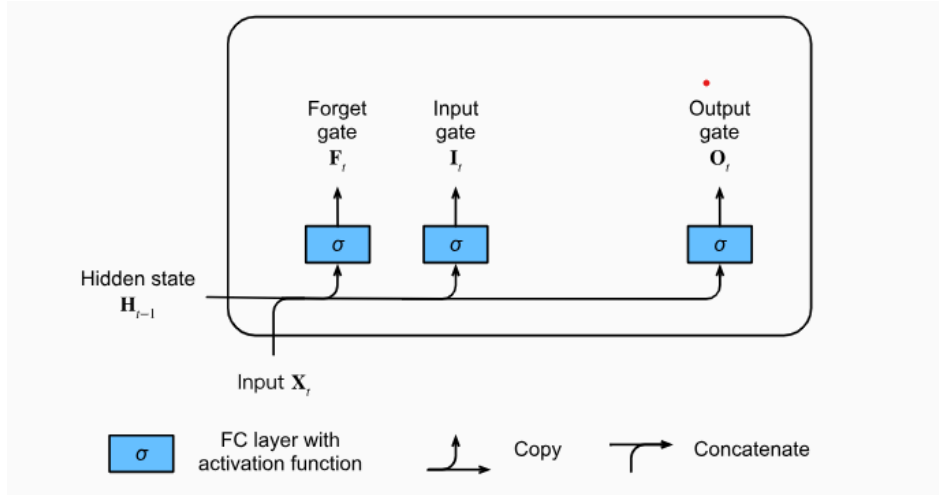


Figura 2.2: Estructura interna de una celda *LSTM*. Imagen obtenida desde [2].

unidimensionales diseñadas de manera causal y, en muchos casos, con dilatación [28, 29], mejorando la eficiencia computacional y reduciendo los problemas de gradientes que suelen afectar a las arquitecturas recurrentes.

En una *convolución causal*, la salida en un instante t depende exclusivamente de valores presentes o pasados de la secuencia de entrada, nunca de valores futuros. Matemáticamente, si se considera una secuencia \mathbf{x} de longitud T y un filtro \mathbf{k} de tamaño K , la salida en el tiempo t se obtiene como:

$$y(t) = \sum_{i=0}^{K-1} k(i) \cdot x(t-i), \quad (2.11)$$

con la condición de que $x(t-i) = 0$ cuando $t-i < 0$. Este mecanismo preserva la causalidad y evita fugas de información desde el futuro hacia el presente.

Para capturar dependencias de mayor alcance sin incrementar excesivamente K , las TCN introducen la *convolución dilatada*, en la que se emplea un factor de dilatación $d \geq 1$. La expresión matemática se modifica a:

$$y(t) = \sum_{i=0}^{K-1} k(i) \cdot x(t-d \cdot i). \quad (2.12)$$

Este ajuste permite expandir el *campo receptivo* de la red sin incrementar significativamente el

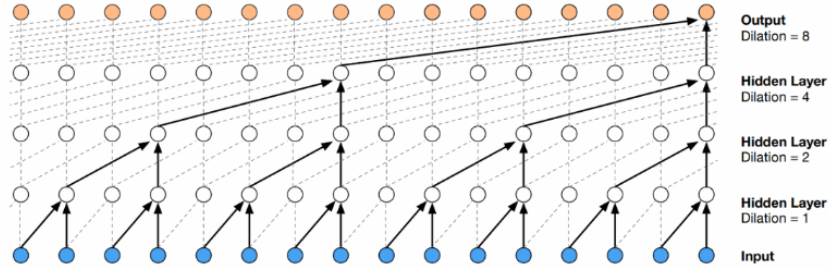


Figura 2.3: Arquitectura típica de una Red Neuronal Convolutiva Temporal. Imagen obtenida desde [3]

número de parámetros. Si las dilataciones crecen de forma exponencial ($d_l = 2^{l-1}$), el campo receptivo total de una pila de L capas con filtros de tamaño K es:

$$R = 1 + \sum_{l=1}^L (K - 1) \cdot d_l. \quad (2.13)$$

Las *TCN* se estructuran habitualmente mediante *bloques residuales*, en los que la salida de un conjunto de capas se suma directamente a su entrada:

$$\mathbf{y} = \mathcal{F}(\mathbf{x}; \theta) + \mathbf{x}, \quad (2.14)$$

funciones de activación no lineales, como *ReLU*. Este diseño facilita el flujo de gradientes, estabiliza el entrenamiento y permite redes más profundas.

En contraste con las redes recurrentes, cuyo modelo básico está dado por:

$$h_t = \phi(W_x x_t + W_h h_{t-1} + b), \quad (2.15)$$

y que requieren propagación secuencial de estados ocultos, las *TCN* reemplazan la recurrencia

por operaciones convolucionales paralelizables. Esto elimina las dependencias estrictamente secuenciales, reduce el riesgo de gradientes desvanecientes o explosivos y mantiene la capacidad de modelar relaciones a largo plazo.

2.3. Estimación híbrida de parámetros mediante ABC-CNN para modelos GARCH-X

El esquema de estimación ABC-CNN para modelos GARCH-X se basa en la generación de para aprender estadísticos resumen altamente informativos pero de baja dimensionalidad. Este enfoque permite aproximar la distribución posterior de los parámetros del modelo sin necesidad de calcular explícitamente la función de verosimilitud. Este algoritmo es descrito en Plaza et.al.(2024, [7]) y esta dado por:

Algoritmo 4 Algoritmo ABC Híbrido para modelo GARCH-X

- 1: **para** $i = 1, \dots, N$ **hacer**
- 2: Proponer $\theta^* = (\omega^*, \alpha^*, \beta^*, \delta^*)^{(i)}$ desde una distribución previa $U(0, 1)$ con la condición:

$$\alpha^* + \beta^* < 1 \quad \text{y} \quad 3\alpha^{*2} + 2\alpha^*\beta^* + \beta^{*2} < 1;$$

- 3: Generar una realización de $X_t^{(i)}$ usando $\theta^{(i)}$
- 4: **fin para**
- 5: Estimar los parámetros del estadístico resumen ($S(\cdot)$) entrenando la red neuronal convolucional (CNN) con la base de datos generada en el Paso 1

$$\left\{ X_t^{(i)} \right\}_{1 \leq i \leq N} \text{ como entradas, y } \left\{ \left(\theta^{(i)} \right) \right\}_{1 \leq i \leq N} \text{ como salidas}$$

del modelo de la CNN.

- 6: Extraer un subconjunto suficientemente grande ($n \geq 0,3 \cdot N$) que no se haya utilizado en el entrenamiento.
 - 7: **para** $i = 1, \dots, N$ **hacer**
 - 8: **repetir**
 - 9: Muestrear θ^* desde la distribución previa $\pi(\theta)$;
 - 10: Generar X' del modelo M dado θ^* ;
 - 11: **hasta que** $\|S(X') - S(X_{\text{obs}})\| < \varepsilon$, usando S como estadístico resumen basado en la CNN
 - 12: Aceptar θ^* y guardar la muestra $\theta^{(i)} = \theta^*$
 - 13: **fin para**
 - 14: Definir $\theta^* = \hat{\theta} = (\hat{\omega}, \hat{\alpha}, \hat{\beta}, \hat{\delta})$
 - 15: Evaluar los resultados observados y estimados con las métricas de desempeño habituales.
-

Capítulo 3

Estimación híbrida de parámetros para modelo GARCH-X

Este capítulo presenta una extensión metodológica del enfoque de estimación de parámetros para modelos *GARCH* con variables exógenas, desarrollada originalmente a través de una combinación de *Approximate Bayesian Computation* (ABC) y redes neuronales convolucionales (*CNN*), en el trabajo realizado por Plaza et. al. (2024 [7]), el esquema híbrido demostró ser efectivo para la estimación en contextos donde la función de verosimilitud es intratable o costosa de evaluar, permitiendo aprender automáticamente estadísticos resumen relevantes a partir de simulaciones del modelo.

El desempeño del enfoque *ABC* depende críticamente de la calidad de los estadísticos resumen utilizados. Aunque las *CNNs* permiten automatizar su construcción, siguen enfrentando desafíos para capturar estructuras secuenciales profundas, especialmente en procesos con alta complejidad dinámica y no linealidad temporal, como los que caracterizan a los modelos GARCH-X con covariables ambientales.

Es por eso, que en este capítulo se propone una extender el esquema híbrido ABC-CNN, comparando tanto el uso de *CNNs* como en el trabajo original, como el uso de redes neuronales recurrentes con memoria a largo plazo, y redes neuronales convolucionales con mecanismos de atención, que han demostrado ser efectivas para capturar dependencias

temporales complejas en datos secuenciales. El objetivo es evaluar si estas arquitecturas pueden mejorar la captura de características temporales relevantes para la estimación de parámetros en modelos GARCH-X, y comparar su desempeño. Luego de elegir la arquitectura con mejor performance, el estudio de simulación se complementa con un benchmark de diferentes métodos de Aproximación Bayesiana Computacional.

3.1. Redes Neuronales utilizadas como estadístico de Resumen

En esta sección se describen las arquitecturas de redes neuronales utilizadas para extraer estadísticos resumen de las series temporales simuladas bajo el modelo *GARCH-X*. A continuación se procede a describir las arquitecturas y entrenamiento de las redes neuronales utilizadas.

3.1.1. Red Neuronal Convolutiva N°1

La primera red neuronal utilizada es una red neuronal convolutiva unidimensional (CNN 1D), diseñada para procesar secuencias temporales. Esta arquitectura se basa en el estudio de Plaza et. al. (2024 [7]), para que el estudio sea comparable.

La arquitectura corresponde a una red neuronal convolutiva profunda unidimensional diseñada para procesar secuencias de 300 pasos temporales con 3 canales de entrada. Inicia con una capa Conv1D de 1024 filtros y tamaño de kernel reducido, seguida de MaxPooling1D para disminuir la resolución temporal. Este patrón de convolución y agrupamiento se repite seis veces, manteniendo siempre 1024 filtros por capa convolutiva, lo que permite extraer representaciones jerárquicas cada vez más abstractas de la señal de entrada mientras se reduce su dimensión temporal de 300 a 3 pasos.

Después de la última capa de pooling, la salida (de forma (3, 1024)) se aplana en un vector de 3072 valores, al que se aplica Dropout para regularización. Luego, el modelo pasa

por una pila de capas densas: primero una de 128 neuronas, seguida de varias capas ocultas de 256 neuronas cada una, con activaciones ReLU y un Dropout adicional antes de la capa de salida. Finalmente, la capa Dense de 4 neuronas produce la predicción final, adecuada, ya que se necesita predecir 4 parámetros del modelo GARCH-X. En total, la red contiene aproximadamente 49 millones de parámetros entrenables (186.94 MB). (Ver detalle en A.1).

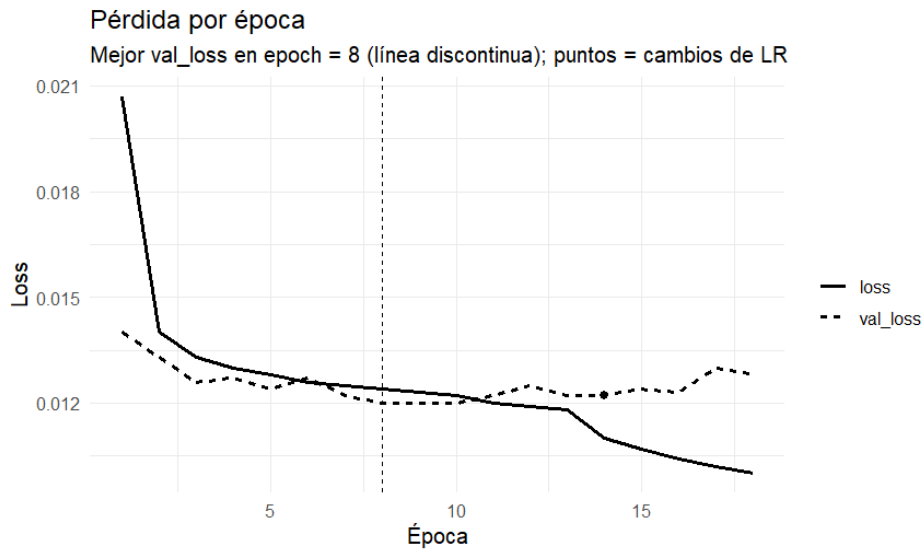


Figura 3.1: Historia de la pérdida de entrenamiento y validación de la CNN 1D.

- **Convergencia inicial:** En la época 1, la pérdida/mse de entrenamiento fue 0,0207 y la validación 0,0140, guardándose el primer modelo debido al *Callback*.
- Se siguió mejorando y se guardaron *checkpoints* en las épocas 2, 3, 5, 7 y 8, alcanzando el mejor `val_loss` = 0,01201 en la época 8.
- Tras la época 8, la validación osciló en el rango 0,0120–0,0127. La tasa de aprendizaje bajó de $1e^{-3}$ a $5e^{-4}$ en la época 13 y luego a $2,5e^{-4}$ en la época 18.
- **Early stopping:** Al no mejorar la validación tras la bajada de *LR*, se disparó el *early stopping* en la época 18, restaurando los pesos de la época 8.

La pérdida de entrenamiento descendió de 0.0207 a 0.0100, mientras la validación

mejoró hasta 0.01201 y luego se estabilizó. El tiempo por época se mantuvo alrededor de $\sim 360\text{--}395$ s ($\approx 6\text{--}6,5$ min), consistente con un modelo grande y lotes masivos de datos.

3.1.2. Red Neuronal CNN + LSTM N°2

Esta arquitectura combina capas convolucionales y LSTM para capturar tanto patrones locales como dependencias temporales a largo plazo en las series temporales.

Comienza con cuatro bloques convolucionales (Conv1D) que incrementan progresivamente la cantidad de filtros (128, 256, 256, 512) para capturar características de distinto nivel de abstracción en la serie temporal. Cada bloque incluye normalización por lotes (*Batch Normalization*) para estabilizar el entrenamiento y una capa de *MaxPooling1D* que reduce la resolución temporal, pasando de 300 observaciones a solo 18.

La salida de la parte convolucional alimenta dos capas LSTM: la primera produce secuencias de 18 pasos con 128 unidades cada uno, y la segunda resume esa información en un único vector de 128 dimensiones. Entre ambas se aplica Dropout para mitigar el sobreajuste.

Luego, la red pasa por una serie de capas densas intercaladas con Batch Normalization y Dropout (256, 128, 64 neuronas), lo que permite refinar la representación aprendida y mejorar la generalización. Finalmente, la capa de salida (Dense) tiene 4 neuronas, adecuadas para un problema de regresión multivariable con 4 que corresponden a los respectivos estadísticos de resumen de parámetros del modelo GARCH-X.

En total, el modelo cuenta con $\approx 1,62$ millones de parámetros entrenables (6,18 MB) y 3200 parámetros no entrenables (12,50 KB), lo que lo convierte en una arquitectura de tamaño moderado pero con suficiente capacidad para manejar patrones temporales complejos.

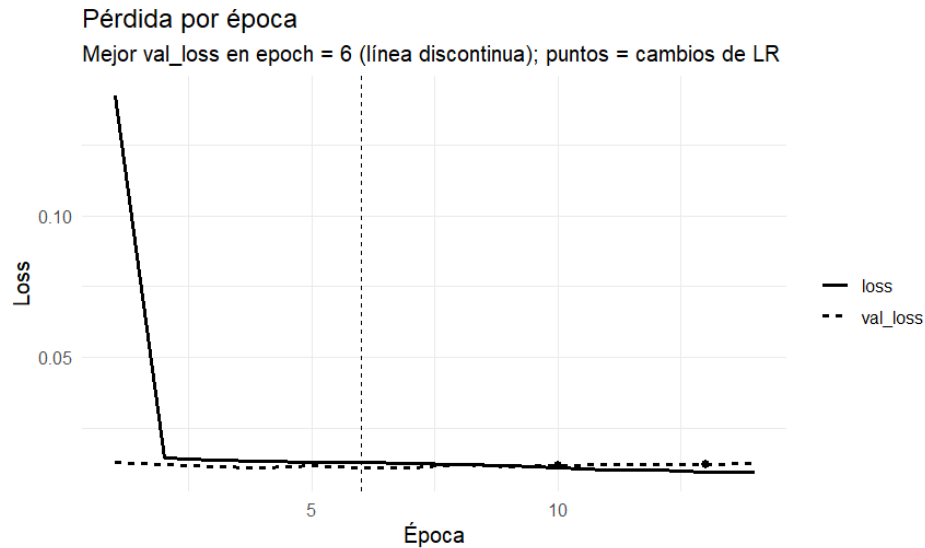


Figura 3.2: Historia de la pérdida de entrenamiento y validación de la CNN + LSTM.

- **Convergencia inicial:** En la época 1, la pérdida/mse de entrenamiento fue 0,1426 y la validación 0,01294, guardándose el primer modelo debido al *Callback*.
- Se siguió mejorando y se guardaron *checkpoints* en las épocas 2, 3, 4 y 6, alcanzando el mejor `val_loss` = 0,01088 en la época 6.
- Tras la época 6, la validación osciló en el rango 0,0109–0,0126. La tasa de aprendizaje bajó de $1e^{-4}$ a $5e^{-5}$ en la época 9 y luego a $2,5e^{-5}$ en la época 12.
- **Early stopping:** Al no mejorar la validación tras las reducciones de *LR*, se activó el *early stopping* en la época 14, restaurando los pesos de la época 6.

3.1.3. Red Neuronal Convolutiva N°3

En esta ocasión se utiliza una arquitectura similar a la red neuronal de la sección 3.1.1, se utiliza una función de pérdida diferente ya que, se utiliza el error cuadrático medio (MSE) ponderado, dándole cinco veces más importancia a la predicción del parámetro β , que captura la persistencia de la volatilidad.

La estructura comienza con una capa Conv1D de gran capacidad (1024 filtros) y kernel pequeño, seguida de una MaxPooling1D para reducir la resolución temporal. Este patrón se repite con variaciones en el tamaño del kernel (4, 12, 12, 6, 6, 6) en seis capas convolucionales adicionales, siempre con 1024 filtros, intercaladas con operaciones de `max pooling` que van reduciendo progresivamente la longitud temporal de la señal. El bloque convolucional finaliza con un aplanado (Flatten) de la representación resultante.

Después de la parte convolucional, la red incluye una densa de 128 neuronas y un bloque de seis capas densas de 256 neuronas cada una, todas con activación ReLU. Se aplican dos capas de Dropout (0,25) para regularizar y reducir el sobreajuste, una después del flatten y otra antes de la capa final. Ver detalle en A.3.

La capa de salida es una Dense lineal con 4 neuronas, correspondiente a las cuatro variables objetivo. La compilación usa el optimizador Adam y la métrica MSE, pero la pérdida es un MSE ponderado (`weighted_mse`) que multiplica el error cuadrático de la tercera salida (volatilidad) por un factor de 5.

En conjunto, es una red muy grande (cerca de 49 millones de parámetros entrenables) (186,94 MB) y 0 parámetros no entrenables.

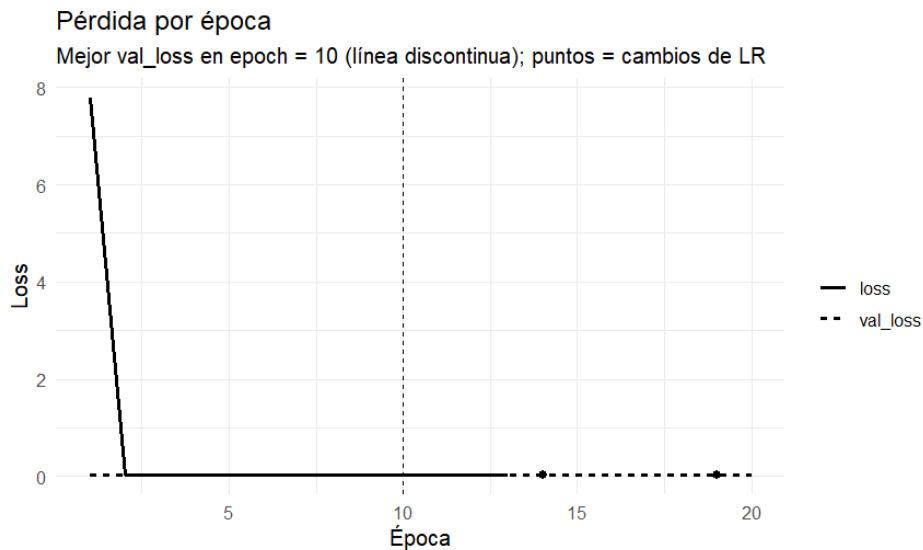


Figura 3.3: Historia de la pérdida de entrenamiento y validación de la CNN 1D con MSE ponderado.

- **Convergencia inicial:** En la época 1, la pérdida/mse de entrenamiento fue 7,7888/1,9529 y la validación 0,03181/0,01595, guardándose el primer modelo debido al *Callback*.
- Se siguió mejorando y se guardaron *checkpoints* en las épocas 2, 3, 6, 7, 8, 9 y 10, alcanzando el mejor `val_loss = 0,02530` en la época 10.
- Tras la época 10, la validación osciló en el rango 0,0253–0,0268. La tasa de aprendizaje bajó de $1e^{-3}$ a $5e^{-4}$ en la época 14 y luego a $2,5e^{-4}$ en la época 19.
- **Early stopping:** Al no mejorar la validación tras las reducciones de *LR* y aparecer valores NaN en la pérdida de entrenamiento a partir de la época 14, se activó el *early stopping* en la época 20, restaurando los pesos de la época 10.

3.1.4. TCN + LSTM N°4

En esta arquitectura se utiliza una red neuronal convolucional temporal (TCN) junto con capas LSTM para capturar tanto patrones locales como dependencias temporales a largo plazo en las series temporales.

3.1.4.1. Descripción de la Arquitectura y Entrenamiento

La arquitectura definida combina convoluciones causales profundas, redes recurrentes y mecanismos de atención para modelar secuencias temporales multivariadas, que en este caso corresponden a la simulación de los retornos, la variable exógena y los retornos al cuadrado.

La red comienza con una entrada de forma $(T, 3)$, donde T es la longitud de la secuencia y 3 los canales de entrada.

TCN + LSTM

Model: "LSTM_TCN"

Layer (type)	Param #	Output Shape
input_series (Input)		(None, T, 3)
temporal_block_1/conv1 (Conv1D)	0	(None, T, 512)
temporal_block_1/bn1 (BatchNorm)	5,120	(None, T, 512)
temporal_block_1/drop (SpatialDropout1D)	2,048	(None, T, 512)
temporal_block_1/conv2 (Conv1D)	0	(None, T, 512)
temporal_block_1/bn2 (BatchNorm)	786,944	(None, T, 512)
temporal_block_1/skip (Conv1D 1x1)	2,048	(None, T, 512)
temporal_block_1/add (Add)	0	(None, T, 512)
temporal_block_2 (same estructura)	1,577,984	(None, T, 512)
temporal_block_3 (same estructura)	1,577,984	(None, T, 512)
temporal_block_4 (same estructura)	1,577,984	(None, T, 512)
temporal_block_5 (same estructura)	1,577,984	(None, T, 512)
temporal_block_6 (same estructura)	1,577,984	(None, T, 512)
lstm_1 (LSTM, 512, return_sequences)		(None, T, 512)

	2,099,200	
dropout_1 (Dropout)		(None, T, 512)
	0	
lstm_2 (LSTM, 512, return_sequences)		(None, T, 512)
	2,099,200	
dropout_2 (Dropout)		(None, T, 512)
	0	
attn/score (Dense->Softmax)		(None, T, 1)
	513	
attn/context (Multiply)		(None, T, 512)
	0	
attn/sum (Lambda sum over time)		(None, 512)
	0	
dense_512 (Dense, relu)		(None, 512)
	262,656	
bn_512 (BatchNormalization)		(None, 512)
	2,048	
drop_512 (Dropout 0.4)		(None, 512)
	0	
dense_256 (Dense, relu)		(None, 256)
	131,328	
bn_256 (BatchNormalization)		(None, 256)
	1,024	
drop_256 (Dropout 0.3)		(None, 256)
	0	
dense_128 (Dense, relu)		(None, 128)
	32,896	
bn_128 (BatchNormalization)		(None, 128)
	512	
drop_128 (Dropout 0.3)		(None, 128)

```

                                0
dense_64 (Dense, relu)          (None, 64)
                                8,256
bn_64 (BatchNormalization)     (None, 64)
                                256
drop_64 (Dropout 0.3)         (None, 64)
                                0
params (Dense, linear, n_params=4) (None, 4)
                                260
=====
Trainable params: ~13,312,069
Non-trainable params : ~14,208
-----

```

En la etapa inicial, la entrada —una serie temporal de tres canales— pasa por seis bloques TCN (Temporal Convolutional Network) con convoluciones causales 1D, normalización por lotes y dropout espacial, con tasas de dilatación crecientes (1, 2, 4, 8, 16, 32) que permiten capturar dependencias de corto y largo plazo sin perder la estructura temporal.

Posteriormente, las salidas de la TCN se procesan mediante dos capas LSTM apiladas de 512 unidades cada una, ambas con retorno de secuencias y seguidas de dropout, para capturar relaciones dinámicas no lineales y dependencias a largo plazo.

A continuación, un bloque de atención temporal calcula ponderaciones sobre cada instante de la secuencia y genera un contexto agregado, lo que concentra la representación en los momentos más relevantes para la predicción.

La etapa final consiste en una serie de capas densas totalmente conectadas (512, 256, 128 y 64 unidades) con activación ReLU, normalización por lotes y dropout decreciente, que proyectan la representación extraída hacia el espacio de salida. La capa de salida es una Dense lineal con tantas neuronas como parámetros objetivo, empleando una pérdida MSE ponderada

que otorga cinco veces más peso al error en la estimación del parámetro β (volatilidad).

Este entrenamiento se realizó con un tamaño de lote de 32, y demoró aproximadamente 9 horas, donde también se utilizaron callbacks, como el *early stopping* y el *model checkpointing*, para evitar el sobreajuste y guardar el mejor modelo durante el entrenamiento.

3.1.5. TCN + LSTM N°5

La arquitectura corresponde a un modelo híbrido que combina bloques convolucionales con conexiones residuales, capas LSTM bidireccionales y un mecanismo de atención temporal, seguido de un conjunto de capas densas totalmente conectadas.

El modelo recibe como entrada secuencias de 300 pasos temporales con tres canales, que son procesadas mediante seis bloques convolucionales de 512 filtros, cada uno compuesto por dos capas Conv1D con activación ReLU, normalización por lotes y spatial dropout, más un atajo residual para preservar la información original. Posteriormente, la salida pasa por dos capas LSTM de 512 unidades (con retorno de secuencias), intercaladas con dropout para regularización.

La representación resultante se somete a un mecanismo de atención temporal que pondera la contribución de cada instante, obteniendo un vector resumen de 512 características. Este vector se procesa en una secuencia de capas densas con 512, 256, 128 y 64 neuronas (todas con ReLU, batch normalization y dropout decreciente), finalizando en una capa de salida Dense lineal con 4 neuronas que representan las variables objetivo.

- **Convergencia inicial:** En la época 1, la pérdida/mse de entrenamiento fue 0,2547 y la validación 0,02602, guardándose el primer modelo debido al *Callback*.
- Se siguió mejorando y se guardaron *checkpoints* en las épocas 2, 3, 4, 5 y 6, alcanzando el mejor `val_loss` = 0,02243 en la época 6.
- Tras la época 6, la validación osciló en el rango 0,0224–0,0271. La tasa de aprendizaje bajó de $1e^{-4}$ a $5e^{-5}$ en la época 11 y luego a $2,5e^{-5}$ en la época 16.

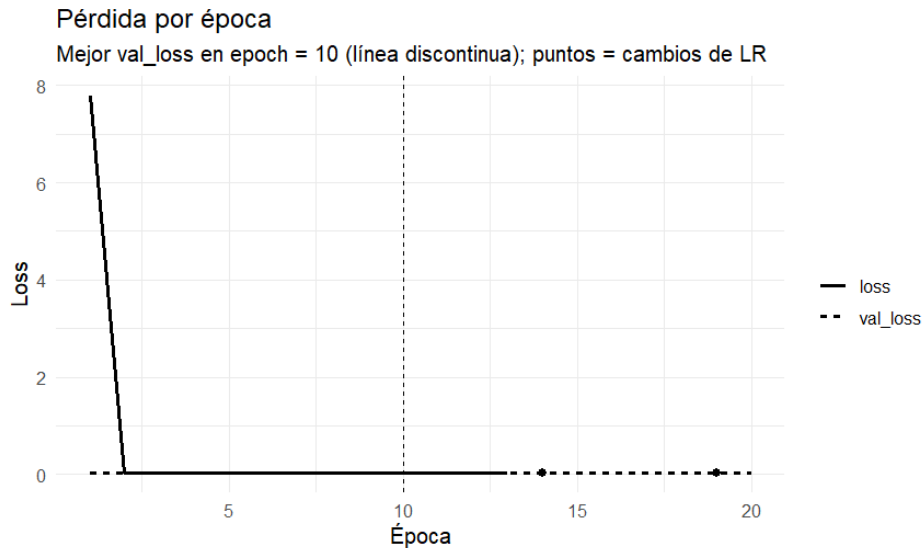


Figura 3.4: Historia de la pérdida de entrenamiento y validación de la TCN + LSTM.

- **Early stopping:** Al no mejorar la validación tras las reducciones de LR , se activó el *early stopping* en la época 16, restaurando los pesos de la época 6.

3.2. Comparativa de rendimiento de Estadísticos de Resumen vía DNN para ABC-Rejection

En esta sección se presenta un benchmark de diferentes arquitecturas de redes neuronales profundas (DNN) utilizadas como estadísticos de resumen para el enfoque de Aproximación Bayesiana Computacional (ABC) en la estimación de parámetros del modelo GARCH-X. El objetivo es evaluar el desempeño de distintas arquitecturas en la captura de características relevantes de las series temporales simuladas, y su capacidad para mejorar la precisión de las estimaciones de parámetros.

La metodología consiste en evaluar varias arquitecturas de DNN, con un Rejection ABC 2, donde se realiza, mil simulaciones de parámetros válidos, tal que se cumplan las siguientes restricciones:

$$\alpha + \beta < 1 \quad \alpha + \beta + \delta < 1 \quad 3 \cdot \alpha^2 + 2\alpha \cdot \beta + \beta^2 < 1$$

Luego se simulan las trayectorias estocásticas del modelo GARCH-X con dichos parámetros, y para cada trayectoria se predicen los estadísticos de resumen con las arquitecturas de *DNN* previamente definidas, y se realiza el ABC Rejection, con trayectorias no utilizadas en el entrenamiento de las *DNN*, como en el esquema propuesto por Plaza et.al (2024 [7]).

3.2.1. Red Neuronal Convolutiva N°1

La primera red neuronal a comparar es una red neuronal convolutiva unidimensional (CNN 1D) descrita en 3.1.1. El estudio arrojó los siguientes resultados:

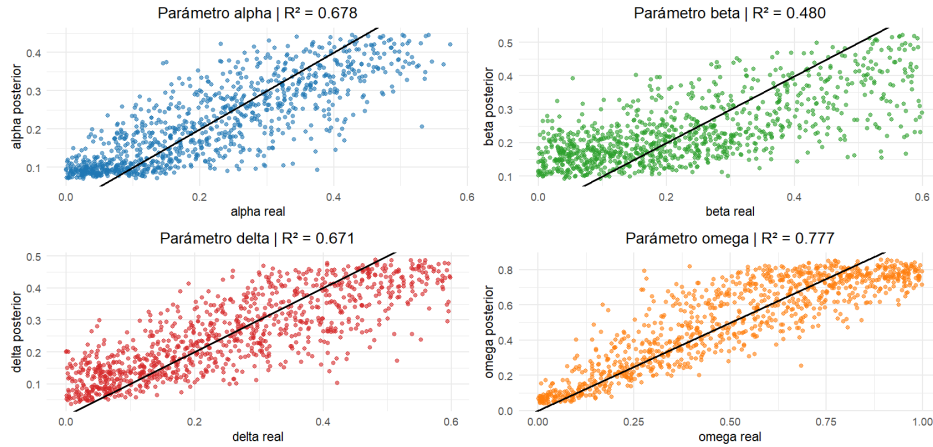


Figura 3.5: Resultados del benchmark de la CNN 1D como estadístico de resumen para ABC-Rejection.

3.2.2. Red Neuronal CNN + LSTM N°2

La segunda red neuronal a comparar es una red neuronal convolutiva unidimensional (CNN 1D) con capas LSTM, descrita en 3.1.2. El estudio arrojó los siguientes resultados:

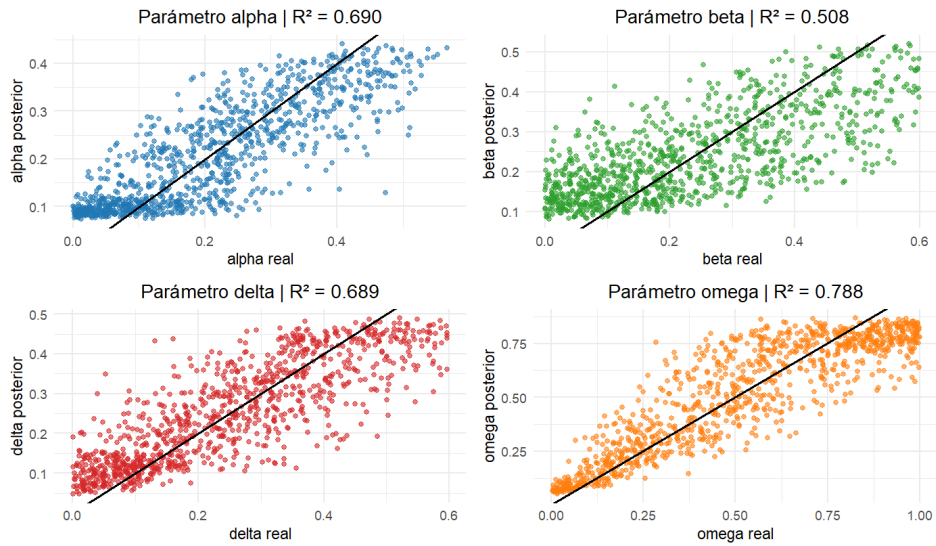


Figura 3.6: Resultados del benchmark de la CNN + LSTM como estadístico de resumen para ABC-Rejection.

3.2.3. Red Neuronal Convolutiva N°3

La tercera red neuronal a comparar es una red neuronal convolutiva unidimensional con función de pérdida MSE ponderado, descrita en A.3. El estudio arrojó los siguientes resultados:

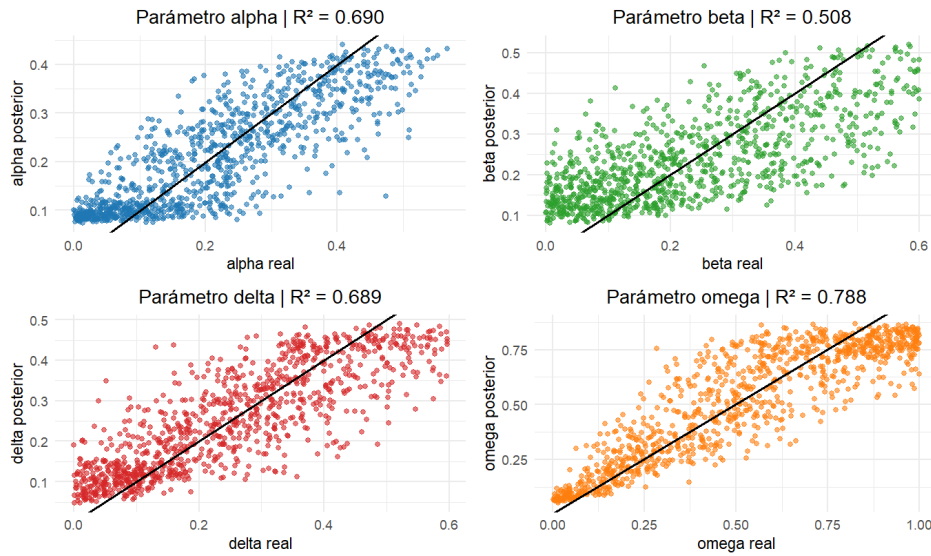


Figura 3.7: Resultados del benchmark de la CNN 1D con MSE ponderado como estadístico de resumen para ABC-Rejection.

3.2.4. TCN + LSTM N°4

La cuarta red neuronal a comparar es una red neuronal convolucional temporal (TCN) con capas LSTM, descrita en 3.1.4. El estudio arrojó los siguientes resultados:



Figura 3.8: Resultados del benchmark de la TCN + LSTM con MSE ponderado como estadístico de resumen para ABC-Rejection.

3.2.5. TCN + LSTM N°5

La quinta red neuronal a comparar es una red neuronal convolucional temporal (TCN) con capas BI - LSTM, también con mecanismo de atención, descrita en 3.1.5. El estudio arrojó los siguientes resultados:

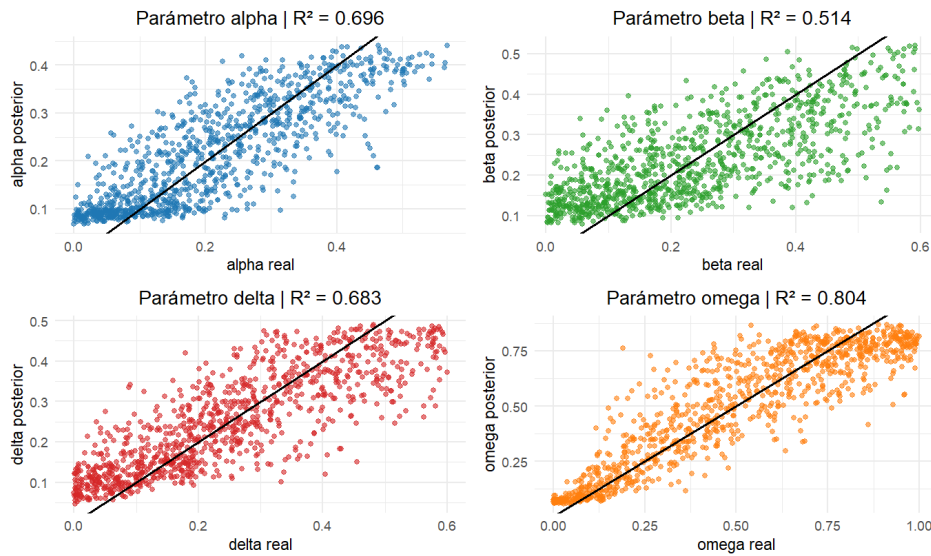


Figura 3.9: Resultados del benchmark de la TCN + LSTM con MSE ponderado como estadístico de resumen para ABC-Rejection.

3.2.6. Comparación de Resultados

Cuadro 3.1: Comparación de R^2 entre parámetros reales y estimados para distintos modelos.

Modelo	ω	α	β	δ
CNN-LSTM	0.7878	0.6899	0.5084	0.6888
CNN	0.7768	0.6782	0.4797	0.6709
CNN-Weighted	0.7719	0.6466	0.4625	0.6342
TCN-LSTM	0.8089	0.7089	0.5478	0.6917
TCN-LSTM-2	0.8039	0.6960	0.5143	0.6835

De esta manera de diferentes métodos de *Aproximación Bayesiana Computacional* teniendo en cuenta el estadístico de resumen que tiene una mejor performance empírica según la tabla 3.1, que en este caso es la arquitectura *TCN + LSTM* N°4, descrita en 3.1.4.

3.3. Comparativa de rendimiento de métodos de Aproximación Bayesiana Computacional

En esta sección se presenta una comparación de diferentes métodos de Aproximación Bayesiana Computacional (*ABC*)

3.3.1. ABC - Rejection

Para la implementación del esquema *ABC-Rejection* se utilizó como medida de distancia, la distancia euclidiana, con una tolerancia ε fija, controlando muchos menos hiperparámetros que en los otros esquemas. Para la version más básica de *ABC-Rejection*, a pesar de su simplicidad, es eficaz sin embargo tiene posterioris más dispersas y menos precisas.

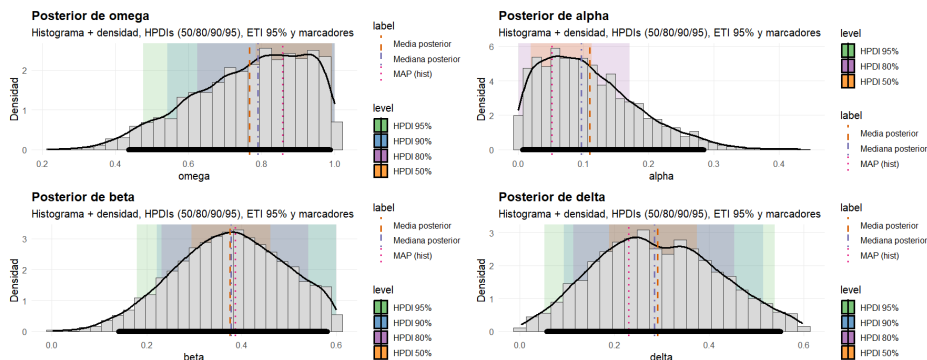


Figura 3.10: Resultados de una sola réplica de ABC-Rejection

al tener menos parametros que calibrar tiene menos margen de mejoras, pero es un buen punto de partida para comparar con los otros esquemas.

3.3.2. ABC - MCMC

Para la implementación del esquema *ABC-MCMC* se utilizó como medida de distancia, la distancia de *Mahalanobis*, la cual es una medida de distancia que tiene en cuenta la correlación entre las variables. Para la implementación de esta medida es necesaria una calibración previa de la matriz de covarianza, la cual se realiza en la fase piloto, junto con la calibración de la tolerancia ε .

Calibración piloto de ε y de la métrica

Para inicializar el esquema *ABC-MCMC* utilizamos una fase de *pilotos*, cuyo objetivo es calibrar tanto la tolerancia ε como la métrica de distancias. Sea X_{obs} la trayectoria observada, y $s_{\text{obs}} = S(X_{\text{obs}})$ el vector de estadísticos resumen extraído mediante la red neuronal (TC+LSTM). A partir de ella se generan n_{pilot} trayectorias sintéticas $X^{(i)} \sim f(\cdot | \theta^{(i)})$, con $\theta^{(i)}$ muestreados de la prior $\pi(\theta)$, y se obtienen los correspondientes resúmenes $s^{(i)} = S(X^{(i)})$.

Estimación de la covarianza.

Con la nube $\{s^{(i)}\}_{i=1}^{n_{\text{pilot}}}$ se estima la media μ y la matriz de covarianza C . Para evitar inestabilidades numéricas se emplea una regularización de cresta Prangle [30] (2017):

$$\Sigma = C + \lambda I, \quad \Sigma^{-1} \approx (C + \lambda I)^{-1}, \quad \lambda \approx 10^{-3}.$$

De este modo se obtiene la precisión robusta requerida para la distancia de Mahalanobis.

Distribución de distancias.

Se calculan las distancias

$$d_i = \|s^{(i)} - s_{\text{obs}}\|_{\Sigma^{-1}} = \sqrt{(s^{(i)} - s_{\text{obs}})^\top \Sigma^{-1} (s^{(i)} - s_{\text{obs}})}.$$

La distribución empírica de $\{d_i\}$ aproxima la variabilidad de los resúmenes simulados alrededor del observado.

Selección de cuantiles.

El umbral ε se fija como un cuantil de dicha distribución:

$$\varepsilon = Q_q(\{d_i\}),$$

donde Q_q denota el cuantil q . Valores altos de q proporcionan tolerancias iniciales más laxas, mientras que valores bajos endurecen el criterio en fases posteriores. Este procedimiento asegura que la cadena *ABC-MCMC* comience con una tasa de aceptación razonable y que progresivamente concentre la posterior alrededor de los parámetros que generan trayectorias consistentes con los datos (?] (2018); Beaumont et al. [31] (2002)).

3.3.2.1. Proceso de calibración de hiperparámetros y estimación

Al implementar el procedimiento de Inferencia Bayesiana Aproximada para estimar los parámetros $\theta = (\omega, \alpha, \beta, \delta)$ de un modelo tipo *GARCH-X*, utilizando resúmenes obtenidos mediante la red neuronal previamente mencionada, donde para ejecutar la respectiva comparación de rendimiento para este método de inferencia aproximada, se definen los siguientes pasos y parámetros para replicar mil veces el experimento.

Generación de trayectorias A partir de una propuesta de parámetros θ , se simula una trayectoria de longitud $n = 300$ tal como fue entrada la red neuronal.

donde $Z_t \sim \mathcal{N}(0, 1)$ y $\varepsilon_t \sim \mathcal{N}(0, 1)$. De la trayectoria se forman tres canales:

$$\{X_t, Z_t, X_t^2\}, \quad t = 1, \dots, n,$$

Distancias y calibración de ε (pilotos) Con la trayectoria observada s_{obs} , se generan $N_{\text{pilot}} = 800$ simulaciones $\{s^{(i)}\}$ con parámetros $\theta^{(i)}$ aleatorios para estimar la matriz de

covarianzas Σ de los resúmenes y su inversa regularizada

$$\Sigma_{\lambda}^{-1} = (\Sigma + \lambda I)^{-1}, \quad \lambda = 10^{-3}.$$

Las distancias se definen como

$$d(s^{(i)}, s_{\text{obs}}) = \sqrt{(s^{(i)} - s_{\text{obs}})^{\top} \Sigma_{\lambda}^{-1} (s^{(i)} - s_{\text{obs}})}.$$

Los cuantiles $q_{\text{lo}} = 0,08$ y $q_{\text{hi}} = 0,45$ determinan umbrales iniciales ($\varepsilon_{\text{lo}}, \varepsilon_{\text{hi}}$) para el esquema adaptativo.

Cadena ABC-MCMC adaptativa La cadena se ejecuta por $N = 14000$ iteraciones, con un período de *burn-in* $B = 3500$. Se utiliza una propuesta uniforme local con reflexión, de tamaño de paso inicial $\delta_0 = 0,045$ adaptada por el esquema de *Robbins-Monro* Ruppert [32] (1988) hacia una tasa de aceptación objetivo $\alpha^* = 0,30$. Mientras que la probabilidad de efectuar propuestas globales (*independence sampler*) se estableció en $p_{\text{global}} = 0,12$.

El criterio de aceptación es:

$$\theta' \text{ aceptado si } d(s(\theta'), s_{\text{obs}}) < \varepsilon_t,$$

donde ε_t decrece adaptativamente desde ε_{hi} hasta ε_{lo} , siguiendo un cronograma exponencial en dos fases:

$$\varepsilon_t = \varepsilon_{\text{hi}} \left(\frac{\varepsilon_{\text{lo}}}{\varepsilon_{\text{hi}}} \right)^{t/\text{adapt_end}}, \quad t \leq \text{adapt_end}, \quad (3.1)$$

$$\varepsilon_t \rightarrow 0,70 \varepsilon_{\text{lo}}, \quad t > \text{adapt_end}. \quad (3.2)$$

. Así obtenemos como resultado la siguientes cadenas para los parámetros del modelo.

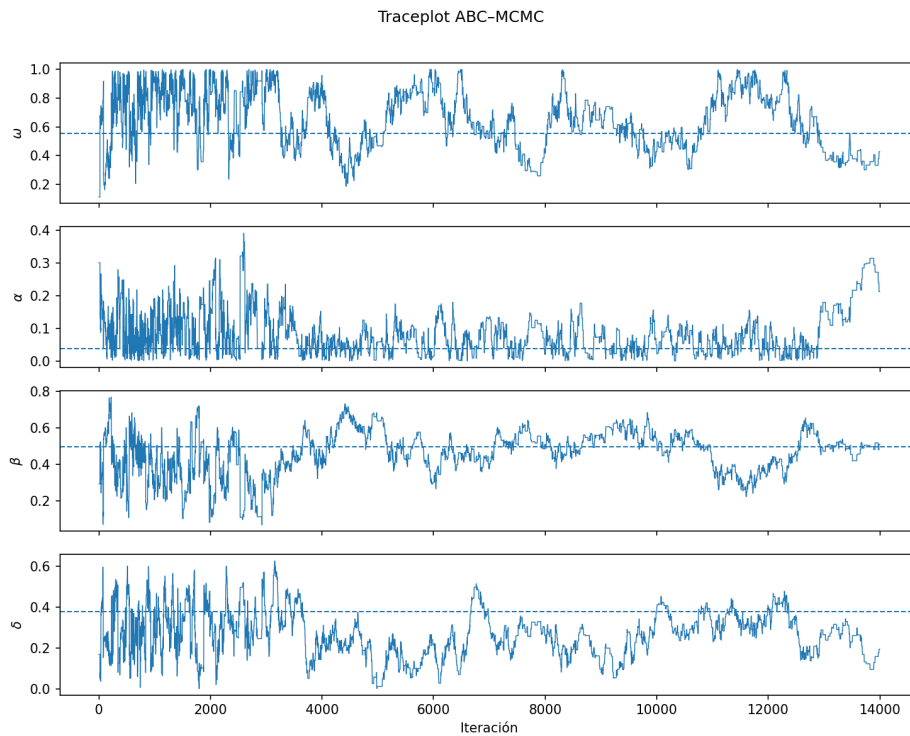


Figura 3.11: Cadenas de Markov generadas por el esquema ABC-MCMC para los parámetros del modelo GARCH-X.

Los traceplots obtenidos para los cuatro parámetros del modelo GARCH-X $\theta = (\omega, \alpha, \beta, \delta)$ muestran un comportamiento diferenciado en términos de convergencia y mezcla. En el caso de ω , la cadena presenta oscilaciones alrededor del valor verdadero de referencia, manteniendo una exploración razonable del espacio parametral, aunque con cierta autocorrelación residual. El parámetro α se concentra mayoritariamente en valores bajos cercanos a cero, con episodios de exploración hacia valores más altos, lo que sugiere una contribución reducida de los choques pasados a la varianza condicional y una mezcla algo lenta. La traza de β evidencia estructuras en bloques con desplazamientos entre distintas mesetas, indicando autocorrelación elevada pero con cobertura adecuada de la región de interés en torno al valor verdadero. Finalmente, el parámetro δ presenta mayor inestabilidad, con periodos prolongados de sesgo hacia abajo y oscilaciones erráticas, lo que lo convierte en el parámetro más difícil de identificar bajo los resúmenes actuales. En conjunto, los resultados

sugieren que el procedimiento ABC-MCMC logra una convergencia razonable θ y donde para δ muestran dinámicas más sensibles y mayor incertidumbre en sus estimaciones.

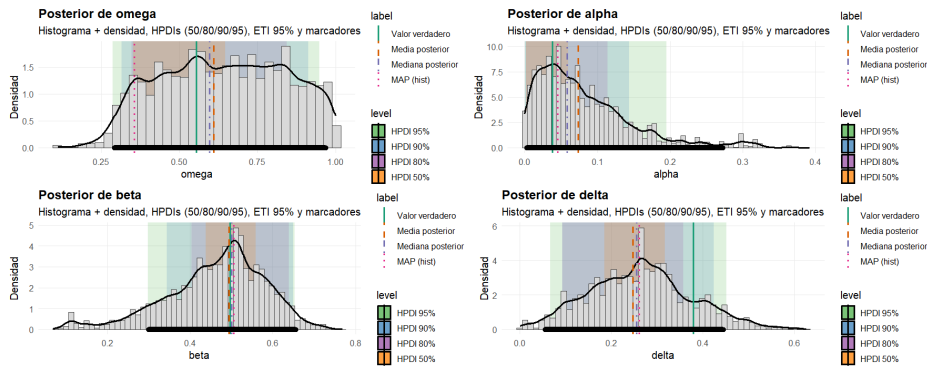


Figura 3.12: Distribuciones posteriores marginales obtenidas por el esquema ABC-MCMC para los parámetros del modelo GARCH-X.

En términos generales, la inferencia ABC-MCMC mostró un buen desempeño en la recuperación de los parámetros. El caso de ω y β resulta satisfactorio, con medias y medianas posteriores muy próximas a los valores verdaderos, y con intervalos HPDI relativamente estrechos que evidencian buena identificabilidad, que en ω a pesar que la media se parece al valor que originalmente genera la trayectoria no logra aproximar a una distribución más informativa. En contraste, α exhibe una distribución posterior más dispersa y sesgada a la izquierda, aunque el valor verdadero se mantiene contenido dentro de los intervalos de credibilidad. Finalmente, δ presenta una subestimación la media posterior se ubica en torno a 0,25, mientras que el valor verdadero corresponde a 0,38, lo que indica mayor dificultad del método para ajustar este parámetro. En conjunto, se observa que los componentes de varianza condicional (ω , β) son más identificables que los asociados a innovaciones (α , δ) para esta réplica.

3.3.2.2. Rendimiento de estimación

Para evaluar el rendimiento del esquema ABC-MCMC, se realizaron mil réplicas independientes del experimento, logrando una dependencia lineal del experimento, tal que:

Comparación	Valor R^2
ω vs ω_{pos}	0.7298
α vs α_{pos}	0.6298
β vs β_{pos}	0.4077
δ vs δ_{pos}	0.6079

Cuadro 3.2: Comparación de R^2 entre parámetros reales y estimados para parámetros de GARCH-X.

Capítulo 4

Aplicación a datos pesqueros del norte de Chile

4.1. Contextualización del problema

El litoral norte de Chile forma parte del sistema de la corriente de Chile-Perú, uno de los ecosistemas marinos más productivos del mundo gracias al afloramiento costero y la advección de nutrientes. En esta zona predomina la pesca pelágica, siendo la anchoveta (*Engraulis ringens*) y la sardina (*Sardinops sagax*) especies clave que, juntas, representaron alrededor del 10 % de las capturas marinas mundiales en 2020.



(a) Sardina (*Sardinops Sagax*).



(b) Anchoveta (*Engraulis Ringens*).

Figura 4.1: Pequeños peces pelágicos.

Su relevancia económica y ecológica hace fundamental su estudio para un aprovechamiento sostenible.

Ambas especies muestran fluctuaciones poblacionales asociadas a condiciones ambientales. En los años 70, el evento El Niño 1972–1973 provocó una fuerte caída en la anchoveta y un aumento en la sardina; a fines de los 80 ocurrió la situación inversa. En general, las sardinas tienden a dominar en periodos cálidos (El Niño), mientras que la anchoveta predomina en condiciones frías (La Niña o regímenes fríos), aunque esta alternancia responde a un conjunto complejo de factores ambientales, climáticos, biológicos y antropogénicos, documentados incluso en registros de más de 250 años(Valdés, 2008 [33]).

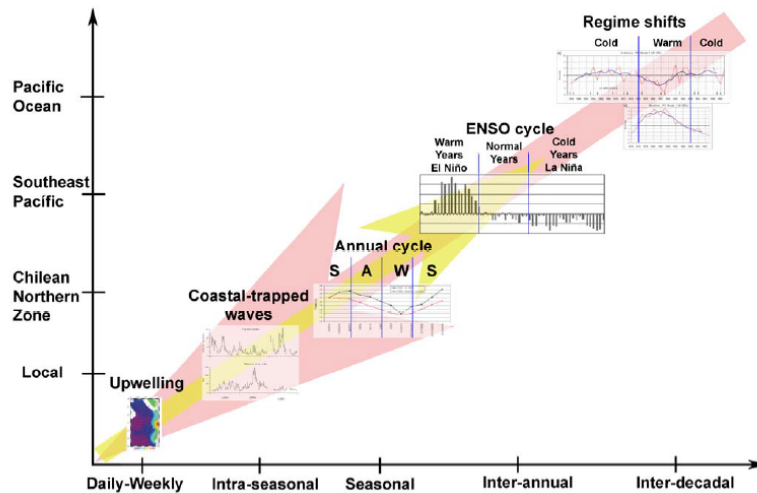


Figura 4.2: Esquema que integra fenómenos locales y globales que influyen en la zona norte de Chile y sus recursos pesqueros, indicando dirección e intensidad de su efecto, obtenido desde (Yáñez, 2008)

La dinámica de estas pesquerías está influida por fenómenos ambientales en diversas escalas:

- **Diaria-semanal:** Cambios en el plancton por variabilidad del afloramiento.
- **Intraestacional:** Desplazamientos de la termoclina ecuatorial que alteran temperatura y nivel del mar.
- **Estacional:** Variaciones de temperatura y termoclina que modifican hábitats y productividad primaria.
- **Interanual:** Eventos ENSO que afectan reproducción, distribución y supervivencia.
- **Interdecadal y multidecadal:** Cambios de régimen (El Viejo y La Vieja) y variabilidad pentadecadal que alternan periodos dominados por sardina o anchoveta.

Un modelo conceptual propuesto por Yáñez (2008 [34]) integra fenómenos locales y globales —como la temperatura superficial del mar, la profundidad de la termoclina y el transporte de Ekman— para explicar cómo estos factores determinan la alternancia de

especies y condicionan su disponibilidad y vulnerabilidad. Comprender estas interacciones es esencial para diseñar estrategias de manejo y esfuerzo ¹ pesquero acordes a la variabilidad ambiental.

4.2. Análisis exploratorio de datos

Una vez entendido el contexto en el cual se enmarca aplicación de la investigación es necesario realizar un análisis exploratorio de los datos, para caracterizar las variable exógenas y las series de desembarques, con el fin de identificar patrones, tendencias, estacionalidades y anomalías que puedan influir en la dinámica de las pesquerías pelagicas en específico de los desembarques de las especies ya mencionadas.

4.2.1. Descripción de los datos

Los datos utilizados en este trabajo corresponden a una recopilación de diversas fuentes de información(SERNAPESCA, 2024[8]; DIRECTEMAR, 2024[9]; NOAA, 2024 [10]; SHOA , 2024 [35]), incluyendo desembarques pesqueros, variables ambientales como temperatura superficial del mar, salinidad, y otros factores relevantes. De esta forma el dataset consolidado tiene 732 observaciones mensuales desde enero de 1963 hasta diciembre de 2023, cuyas variables son:

- *Date*: Fecha de la observación (formato: YYYY-MM-DD).
- *Year*: Año de la observación.
- *Month*: Mes de la observación.
- *LSAR*: Desembarques de sardina (*Sardinops sagax*) en toneladas obtenidas desde el Anuario Estadístico de Pesca y Acuicultura.

¹Cantidad total de actividad pesquera en los caladeros durante un periodo de tiempo determinado

- *LANC*: Desembarques de anchoveta (*Engraulis ringens*) en toneladas obtenidas desde el Anuario Estadístico de Pesca y Acuicultura.
- *SST*: Temperatura superficial del mar en grados Celsius de la estación de Antofagasta.
- *PDO*: Oscilación multidecenal del Pacífico Norte. Cambios de régimen influyen en pesquerías y distribución de especies. – índice climático.
- *SOI*: Índice de Oscilación del Sur. Relacionado con El Niño y La Niña, afecta patrones climáticos y pesquerías.- índice climático.
- *TI*: Índice de turbulencia. Mide la intensidad o grado de irregularidad en el movimiento del agua, generalmente debido a la acción del viento y las olas.
- *MEI*²:Índice multivariado que resume la señal ENSO³ usando múltiples variables (viento, presión, SST, etc.).
- *NI2*: Anomalía mensual de la temperatura superficial del mar (SST, en °C) en la región Niño 1+2, ubicada en el extremo oriental del Pacífico ecuatorial
- *N34*: Anomalía mensual de la temperatura superficial del mar (SST, en °C) en la región Niño 3.4, ubicada en el Pacífico central ecuatorial.
- *HCI*: diferencia de presión atmosférica entre el sistema de alta presión del Pacífico y el sistema de baja presión continental.

4.3. Visualización de las series temporales

Para comenzar el análisis exploratorio de los datos, se procede a visualizar las series temporales de las variables más relevantes. En primer lugar, se grafican los desembarques de

²Desde 1963 hasta 1978 calculado por su versión antigua deprecada el 2011, desde 1979 en adelante, se utiliza su versión actualizada

³.El Niño-Oscilación del Sur"

sardina y anchoveta a lo largo del tiempo, junto con las variables ambientales que podrían influir en su dinámica.

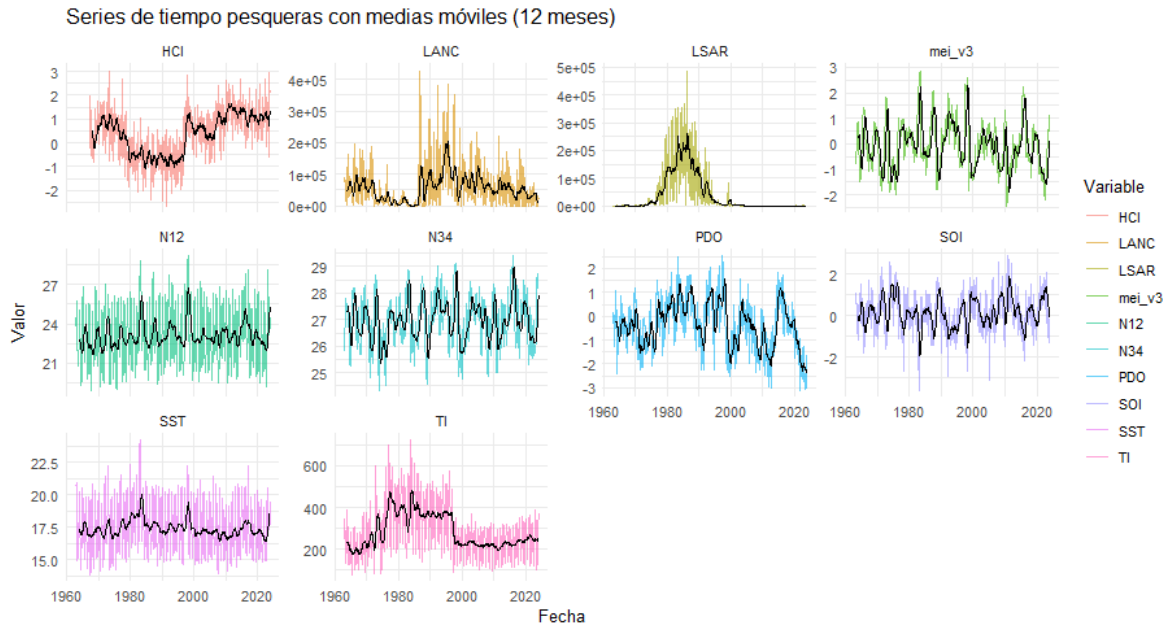


Figura 4.3: Series temporales de desembarques de sardina y anchoveta, junto con variables ambientales relevantes.

La anchoveta (*Engraulis ringens*) y la sardina (*Sardinops sagax*) han mostrado históricamente patrones de alternancia en sus niveles de abundancia, caracterizados por una relación inversa (Shwartzlose, 1996 [?]; Yañez, 2010 [36]; Plaza, 2024 [7])., cuando una de estas especies se encuentra en un periodo de alta disponibilidad, la otra tiende a presentar reducciones poblacionales. Esta dinámica ha sido ampliamente registrada en el contexto del Sistema de Corrientes de Humboldt y se vincula principalmente a la influencia de factores ambientales, especialmente a las fluctuaciones en la temperatura superficial del mar (SST), en distintas escalas temporales. Entre estos factores destacan los eventos El Niño-Oscilación del Sur (ENSO), la propagación de ondas atrapadas a lo largo de la costa, así como los conocidos periodos cálido (“El Viejo”) y frío (“La Vieja”). Esta interacción se puede ver representada,

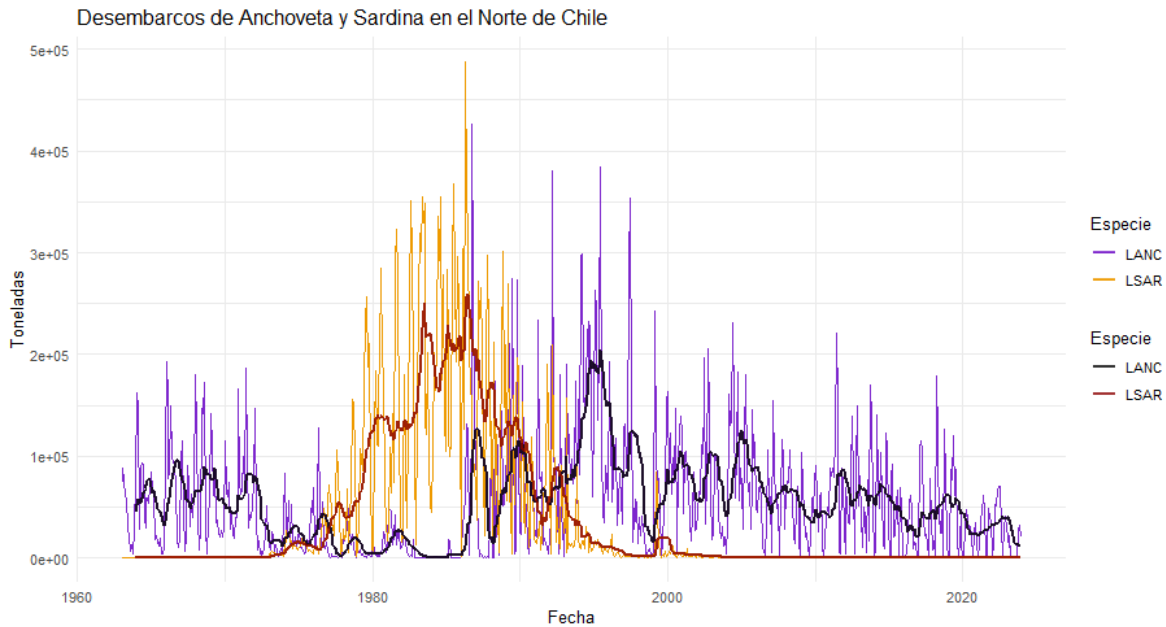


Figura 4.4: Series temporales de desembarques de sardina y anchoveta, y sus medias móviles de 12 meses.

4.3.1. Análisis de retornos

Para evaluar la volatilidad de las series de desembarques, se procede a calcular los retornos y retornos logarítmicos mensuales de las series de interés. Los retornos de la serie de desembarques de Sardina (*Sardinops sagax*), son tal que,

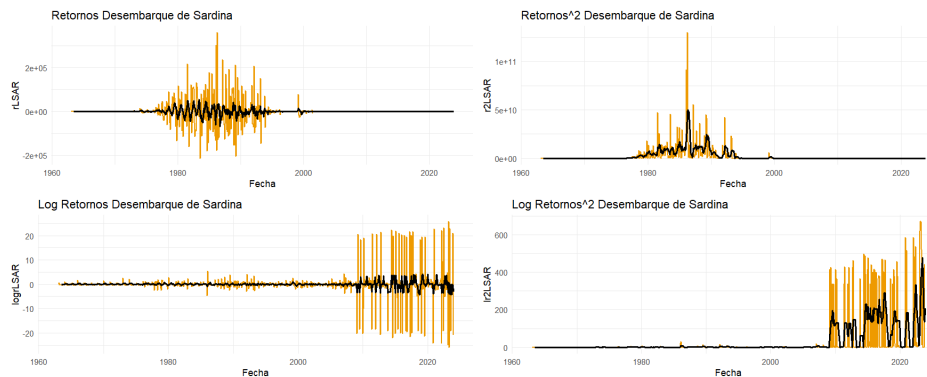


Figura 4.5: Retornos y retornos logarítmicos de las series de desembarques de Sardina.

En contra parte, los retornos de la serie de desembarques de Anchoveta (*Engraulis*

ringens), son tal que,

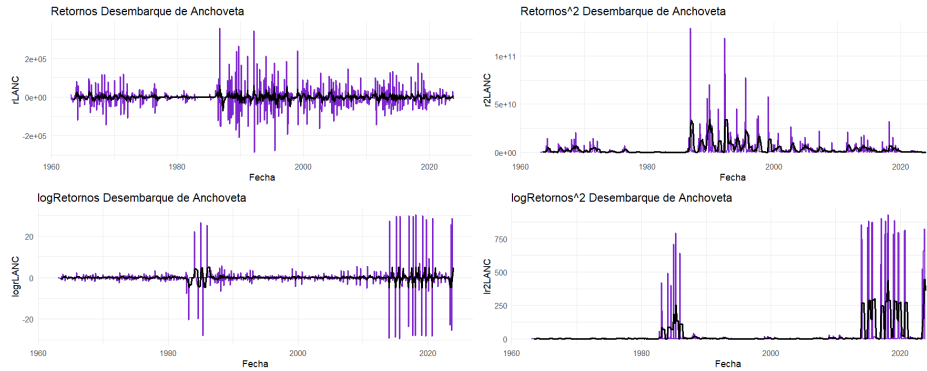


Figura 4.6: Retornos y retornos logarítmicos de las series de desembarques de Anchoveta.

4.3.2. Test de raíz unitaria

Para evaluar la estacionariedad de las series de retorno, se aplica el test de raíz unitaria de Dickey-Fuller aumentado (ADF). Este test es fundamental para determinar si las series son estacionarias. El test ADF se aplica a las siguientes series de retorno

Cuadro 4.1: Resultados del test ADF para las series de retorno de LSAR

Variable	Estadístico ADF	Orden de rezago	Valor-p
r_{LSAR}	-18.783	9	0.01
r^2_{LSAR}	-5.437	9	0.01
$\log(r_{LSAR})$	-12.469	9	0.01
$\log(r^2_{LSAR})$	-5.196	9	0.01

Cuadro 4.2: Resultados del test ADF para las series de retorno de LANC

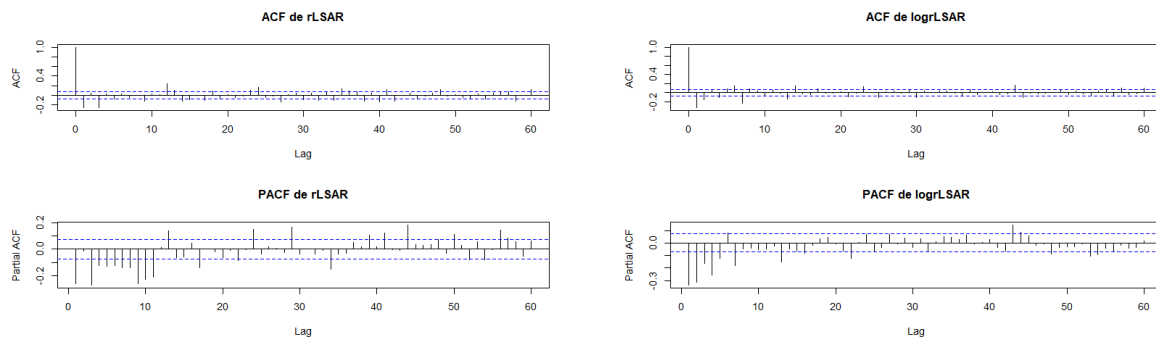
Variable	Estadístico ADF	Orden de rezago	Valor-p
r_{LANC}	-13.982	9	0.01
r^2_{LANC}	-7.495	9	0.01
$\log(r_{LANC})$	-14.220	9	0.01
$\log(r^2_{LANC})$	-6.473	9	0.01

En todos los casos, los valores del estadístico Dickey-Fuller son significativamente menores que los valores críticos al 1% de significancia, y los valores-p son menores a 0,01. Por lo tanto, se rechaza la hipótesis nula de presencia de raíz unitaria para cada serie, concluyéndose que todas las transformaciones de los retornos de los desembarques de Sardina son *estacionarias*.

4.3.3. Análisis de correlación en los retornos

Al realizar un análisis de correlación en los retornos y log-retornos de las series de desembarques, se observa que las series de retornos y log-retornos de Sardina y Anchoveta presentan leves autocorrelaciones significativas, para algunos rezagos en específico, sin embargo, al analizar los retornos al cuadrado y log-retornos al cuadrado, se evidencia una estructura de dependencia temporal marcada por una fuerte autocorrelación en los primeros *rezagos* de las series que decrece lentamente, lo que sugiere que la volatilidad de los retornos es persistente en el tiempo, lo que justifica el uso de modelos GARCH para capturar esta dinámica. Los gráficos de la función de autocorrelación (ACF) y la función de autocorrelación parcial (PACF) muestran que los retornos y log-retornos de las series de desembarques se presentan a continuación, rezagados, lo que sugiere una posible dependencia temporal en los datos.

Figura 4.7: ACF y PACF de los retornos y log-retornos de la Sardina



(a) Retornos r_{LSAR}

(b) Log-retornos $\log(r_{LSAR})$

Figura 4.8: ACF y PACF de los retornos al cuadrado y log-retornos al cuadrado de la Sardina

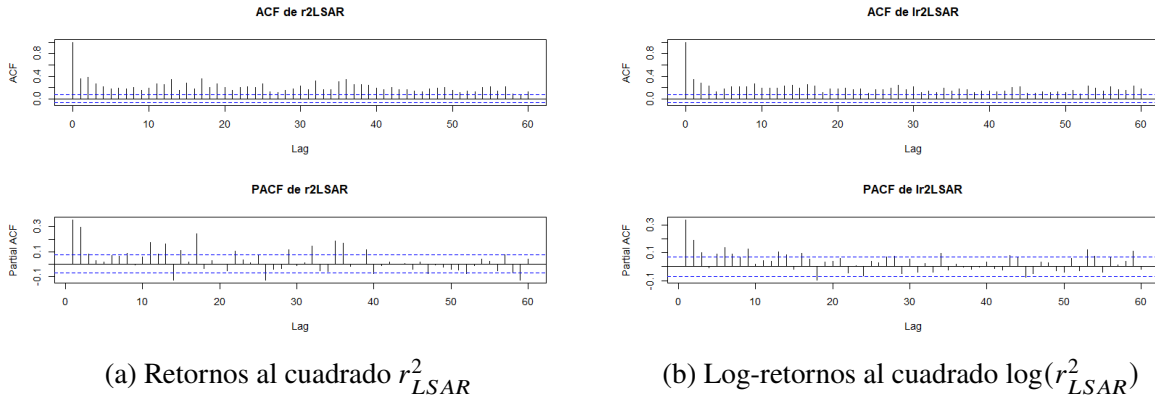


Figura 4.9: ACF y PACF de los retornos y log-retornos de la Anchoveta

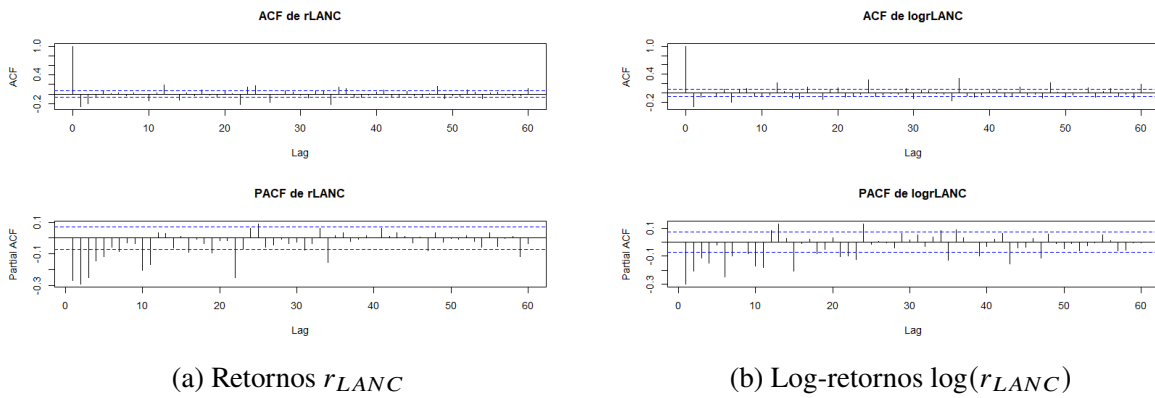
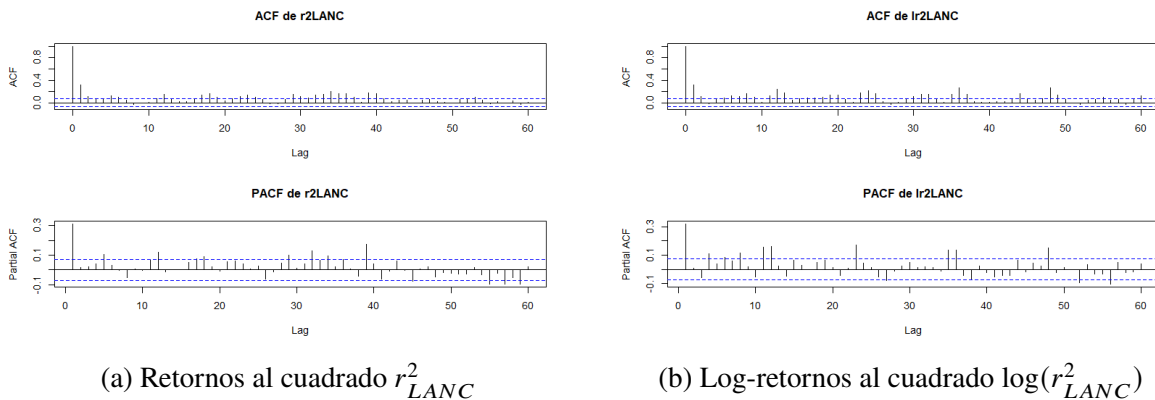


Figura 4.10: ACF y PACF de los retornos al cuadrado y log-retornos al cuadrado de la Anchoveta



Por otro lado los valores de curtosis obtenidos para las series de desembarque, sus retornos y retornos al cuadrado indican una fuerte desviación respecto a la normalidad, especialmente en los retornos al cuadrado, que presentan colas muy pesadas. Esto sugiere la presencia de eventos extremos y variabilidad cambiante en el tiempo, lo que indica la presencia de heterocedasticidad condicional.

Cuadro 4.3: Curtosis de las series de desembarque, retornos y retornos al cuadrado

Variable	Especie	Curtosis
$LSAR$	Sardina	9.39
r_{LSAR}	Sardina	15.24
r_{LSAR}^2	Sardina	98.96
$LANC$	Anchoveta	8.85
r_{LANC}	Anchoveta	9.86
r_{LANC}^2	Anchoveta	67.22

4.3.4. Reducción de dimensionalidad

Para abordar la complejidad de las variables ambientales y su influencia en los desembarques, se propone aplicar técnicas de reducción de dimensionalidad. Estas técnicas permiten identificar patrones y relaciones subyacentes en los datos, facilitando la interpretación y el modelado posterior.

4.3.4.1. Análisis de componentes principales

Al realizar un Análisis de Componentes Principales (PCA) sobre las variables ambientales, se busca extraer las características más relevantes que expliquen la variabilidad en los desembarques de sardina y anchoveta para evaluarlas como variable exógena en el modelo.

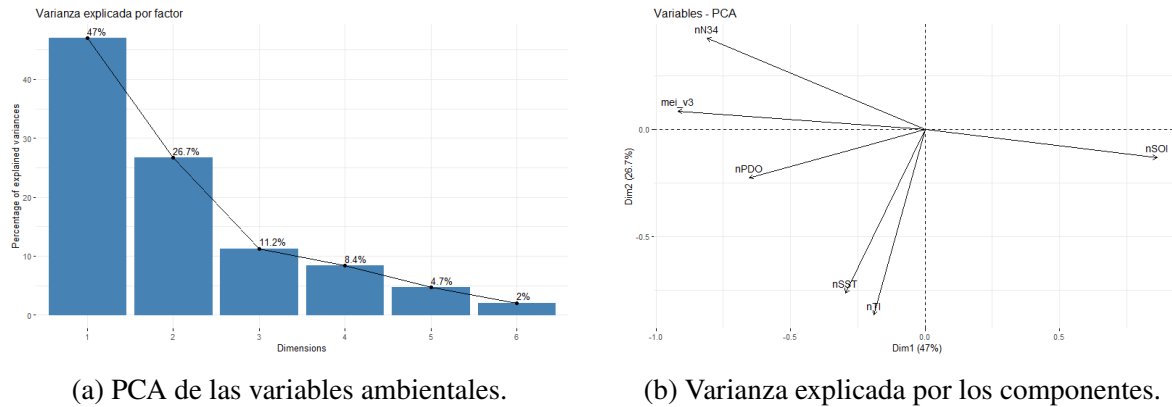


Figura 4.11: Análisis de Componentes Principales (PCA).

Los resultados del PCA muestran que los primeros dos componentes principales explican más del 70 % de la variabilidad, por lo que se puede considerar que estos componentes capturan la mayor parte de la información relevante de las variables ambientales.

El PCA indica que la primera dimensión (47 %) refleja un gradiente térmico entre condiciones cálidas (SST, TI, Niño 3.4, MEI) y frías (SOI), relevante para diferenciar ambientes favorables para anchoveta y sardina. La segunda dimensión (26.7 %) capta variabilidad interanual y decadal (MEI, Niño 3.4, PDO), asociada a efectos ambientales de distinta escala temporal sobre los ecosistemas pesqueros.

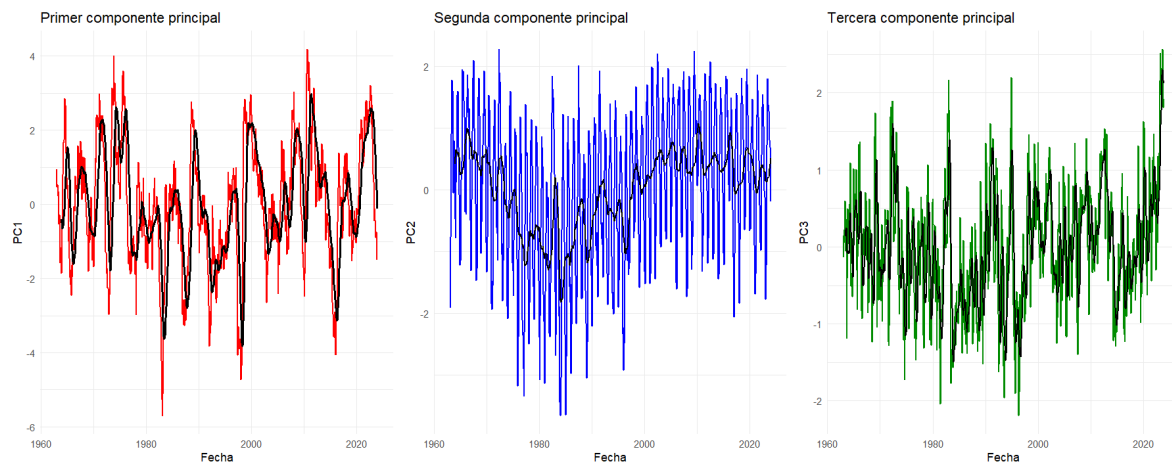
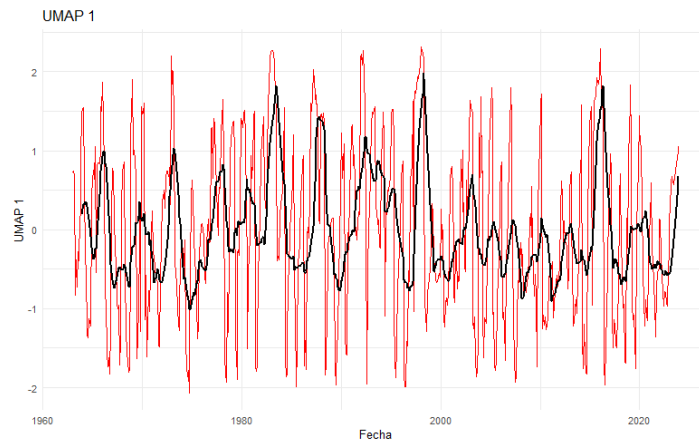


Figura 4.12: Componente principal dos como serie de tiempo.

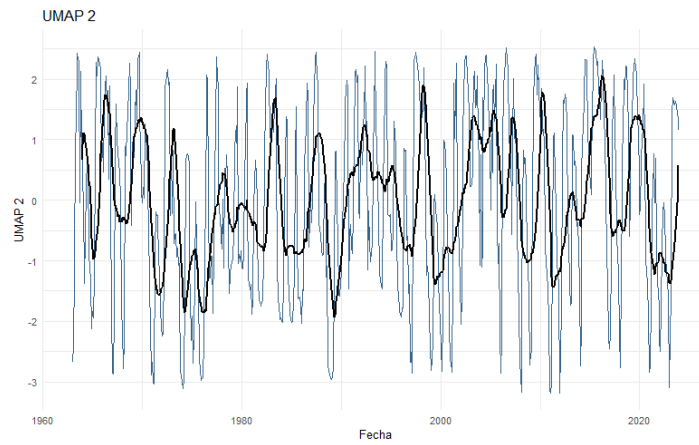
Figura 4.13: Análisis de Componentes Principales como series de tiempo (PCA).

4.3.5. Uniform Manifold Approximation and Projection (UMAP)

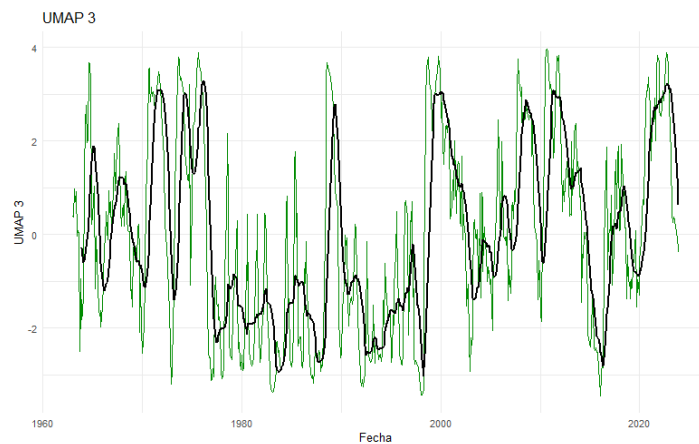
Para capturar la estructura no lineal de los datos, se aplica UMAP (Uniform Manifold Approximation and Projection), una técnica de reducción de dimensionalidad que preserva la topología local y global de los datos. En este caso se ajustó un hiperparámetro de para la reducción a tres dimensiones, cuyas trayectorias son en el tiempo son,



(a) Trayectoria que captura la mayor variación estructural global.



(b) Trayectoria aporta información complementaria a UMAP1.



(c) Trayectoria que captura estructuras locales.

Figura 4.14: Representaciones UMAP de trayectorias transformadas.

De esta forma $UMAP1$, $UMAP2$ y $UMAP3$ son coordenadas latentes que representan una proyección no lineal de los datos originales. Los puntos cercanos en este espacio tienden a ser similares en el espacio original. Cabe destacar que tanto para PCA como para UMAP, los datos fueron estandarizados, para asegurar que todas las variables contribuyan de manera equitativa al análisis.

Con ello se pretende probar la estimación híbrida de parámetros en el modelo GARCH-X, utilizando las componentes principales y las coordenadas UMAP como variables exógenas.

4.4. Estimación híbrida de parámetros aplicada a los desembarques de Sardina y Anchoqueta

Se realiza la experimentación de la metodología propuesta en el capítulo 4, aplicada a los datos de desembarques de sardina y anchoqueta, utilizando las variables exógenas obtenidas mediante el Análisis de Componentes Principales (PCA) y UMAP. Con el fin de seguir extendiendo el estudio realizado por Plaza-Vega and Araya [7], se pone a prueba un método de ABC-MCMC, a pesar de que la precisión en la estimación resultó ser menor.

4.4.1. Resultados para la Anchoqueta

Se evaluaron 14 modelos GARCH-X(1,1,1) con una covariable exógena Z por ajuste (SST, TI, PDO, N12, N34, SOI, HCI, MEI, PC1–PC3, umap1–umap3). Las *tasas de aceptación* totales oscilaron entre 0,245 y 0,356 (mediana en torno a 0,28), mientras que las *aceptaciones dentro del soporte a posteriori* se ubicaron entre 0,202 y 0,347. Los tiempos de cómputo por corrida fueron estables (aprox. 518–522 s).

Parámetro de nivel ω . Las medias a posteriori se concentraron en el rango [0,08, 0,15], con valores relativamente mayores para PC2 ($\bar{\omega} \approx 0,146$), PC1 ($\approx 0,122$) y PDO ($\approx 0,132$). Las distribuciones de ω exhiben *colas anchas* y asimetrías (ETI y HPDI amplios), con masas apreciables cerca de cero en varias covariables (p.ej. SST, TI, umap1, umap3), lo que sugiere

una contribución basal de varianza condicional moderada y con alta incertidumbre.

Parámetro ARCH α . En general se ubicó entre 0,25 y 0,33, con casos más altos en MEI ($\bar{\alpha} \approx 0,408$), PC1 ($\approx 0,319$), TI ($\approx 0,320$) y PDO/N34/umap2 ($\approx 0,28-0,29$). Estos resultados indican *respuesta inmediata* de la varianza a los choques pasada la media, con mayor sensibilidad cuando $Z = \text{MEI}$.

Parámetro GARCH β . La persistencia osciló entre $\approx 0,31$ y $0,50$ para la mayoría de covariables (HCI y umap3 cerca de $0,495$), destacando un *valor atípicamente bajo* para MEI ($\bar{\beta} \approx 0,136$, MAP $\approx 0,019$), consistente con *baja inercia* de la volatilidad cuando se usa MEI como exógena. En contraste, N12, N34, PDO y PC1 presentan β elevadas ($\gtrsim 0,37$), lo que sugiere *alta persistencia* de la varianza condicional.

Carga exógena δ . Las medias se situaron mayormente en $[0,06, 0,10]$, con el mayor aporte en N12 ($\bar{\delta} \approx 0,105$) y valores relativamente altos en PC1/PC2/umap1/umap3. Los intervalos (ETI/HPDI) de δ son amplios y, en varios casos, incluyen valores cercanos a cero, lo que evidencia *heterogeneidad* en la fuerza con que la covariable Z impacta la volatilidad.

En términos de mecanismos, la tríada (α, β, δ) indica que:

- (i) existe una *componente ARCH* no despreciable (choques recientes importan);
- (ii) la *persistencia GARCH* varía por covariable (de moderada a alta), con un caso de *baja inercia* (MEI);
- (iii) la *señal exógena* aporta de forma diversa según Z , siendo N12 y PC1 ejemplos de contribuciones más claras.

La combinación de $\alpha + \beta$ cercana a 1 en varios ajustes sugiere *persistencia elevada*, mientras que las colas pesadas y la asimetría en ω y δ recomiendan robustecer el modelado de errores (p. ej., t -Student/GED) y comparar con estimadores MAP/mediana cuando la media posterior muestre sesgos por colas.

4.4.2. Resultados para la Sardina

Se ajustaron nuevamente 14 modelos GARCH-X(1,1,1) considerando una covariable exógena Z por corrida (SST, TI, PDO, N12, N34, SOI, HCI, MEI, PC1–PC3, umap1–umap3). Las *tasas de aceptación* totales se ubicaron entre 0,216 (N12) y 0,391 (MEI), con mediana cercana a 0,26; las *aceptaciones a posteriori* variaron entre 0,177 (N12) y 0,383 (MEI). Los tiempos de cómputo fueron estables, del orden de 523–543 s por ajuste.

Nivel ω . Las medias a posteriori se concentraron en [0,081, 0,136] (mín. PC1; máx. N12), con ETI/HPDI amplios y masa apreciable cerca de cero en varias covariables (p. ej., SST, TI, N34, PC1–PC2, umap1–umap3), lo que sugiere una contribución basal de varianza condicional moderada y con alta incertidumbre.

ARCH α . En la mayoría de los ajustes α se situó en torno a 0,22–0,31 (SST, TI, PDO, N34, SOI, HCI, PC1–PC3, umap1), con valores *marcadamente superiores* para MEI ($\bar{\alpha} \approx 0,40$) y, especialmente, para umap2 y umap3 ($\bar{\alpha} \approx 0,46$ y 0,45, respectivamente). Estos casos indican una mayor *sensibilidad inmediata* de la varianza ante choques recientes.

GARCH β . La persistencia fue *moderada-alta* en la mayoría de las covariables ($\bar{\beta} \approx 0,40$ –0,54; p. ej., SST, N34, SOI, PC1, PC2, umap1), mientras que fue *notablemente baja* para MEI ($\bar{\beta} \approx 0,11$), umap2 ($\approx 0,09$) y umap3 ($\approx 0,12$). Por tanto, con estas tres covariables la volatilidad muestra menor inercia temporal.

Carga exógena δ . Las medias se concentraron entre 0,07 y 0,12, destacando N12 ($\bar{\delta} \approx 0,123$) y valores relativamente altos en TI, PC2–PC3 y umap1. Los intervalos de credibilidad (ETI/HPDI) son amplios y, en varios casos, incluyen valores cercanos a cero, indicando heterogeneidad en la fuerza del efecto exógeno según Z .

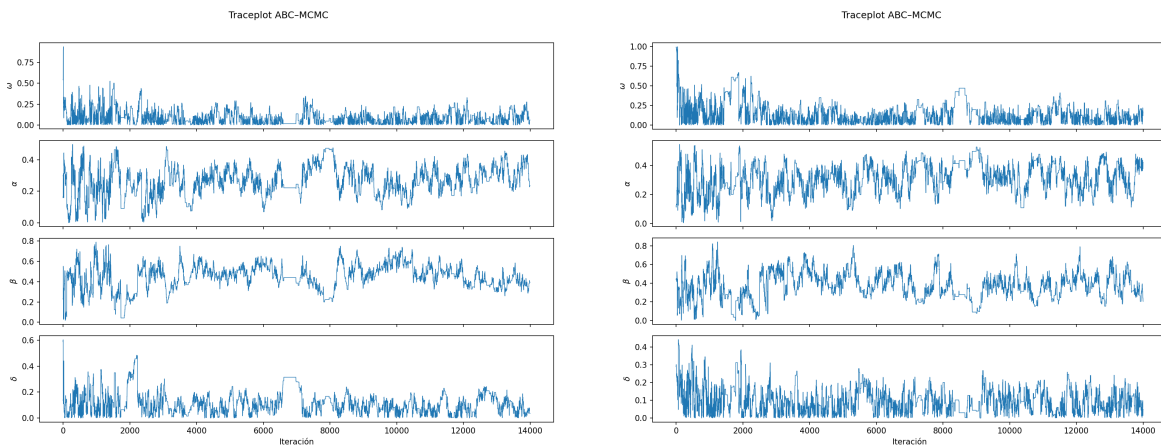
Para la mayoría de las covariables (SST, N34, SOI, PC1–PC3, umap1) se observa el patrón α moderada + β elevada, con $\alpha + \beta$ típicamente en 0,65–0,80, consistente con *alta persistencia* de la volatilidad condicional. En contraste, con MEI, umap2 y umap3 se obtiene α alta + β baja, con $\alpha + \beta \approx 0,52$ – $-0,57$, perfil más *reactivo a choques* y menos persistente. En conjunto, las evidencias indican que la señal exógena mejora el modelado

de la dinámica de varianza de *lsar*, aunque con intensidad y mecanismo (inmediatez vs. persistencia) dependientes de la covariable.

4.4.3. Comparación entre Anchoqueta y Sardina

Para ambas especies, las covariables MEI, *umap2* y *umap3* se asocian a valores elevados de α y bajos de β , lo que sugiere que estas variables exógenas capturan dinámicas de volatilidad más reactivas y menos persistentes. En contraste, covariables como N34, SOI y PC1 tienden a asociarse con β más altas, indicando una mayor persistencia en la volatilidad condicional. Esto sugiere que estas variables exógenas están relacionadas con dinámicas de volatilidad más estables y duraderas. En términos de la carga exógena δ , ambas especies muestran que N12 tiene un impacto relativamente fuerte en la volatilidad, con medias a posteriori más altas en comparación con otras covariables. Sin embargo, la incertidumbre en la estimación de δ es considerable para ambas especies, como lo indican los amplios intervalos de credibilidad.

Así variables como N12 y PC1 parecen tener un impacto más consistente y significativo en la volatilidad de los desembarques de ambas especies.



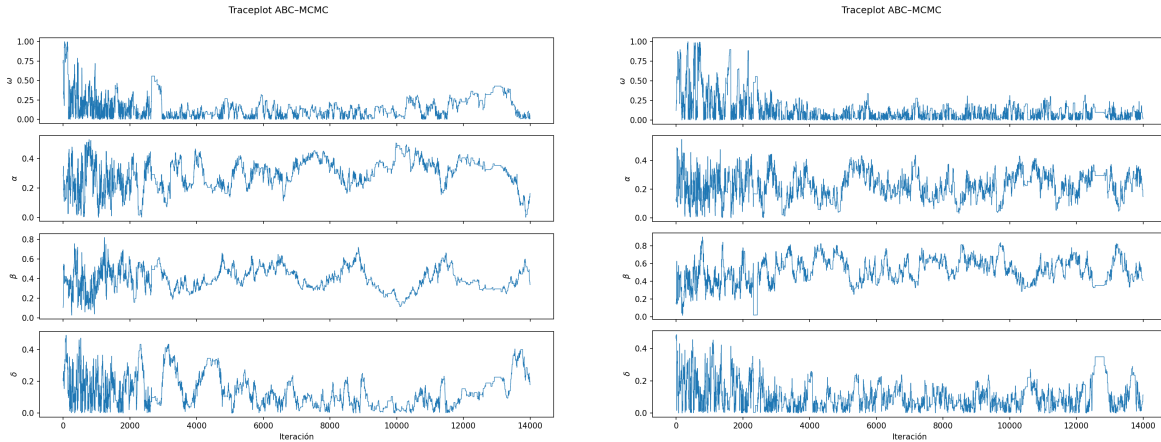
(a) *Traceplot* por parámetro para estimación para la variable Exógena N12.

(b) *Traceplot* por parámetro para estimación para la variable Exógena PC1.

Figura 4.15: Cadenas de las estimaciones para la Anchoqueta.

En la Figura 4.15 se observa que, al usar la covariable N12, las cadenas de los

parámetros son estables y muestran buena mezcla a lo largo de las iteraciones, lo que evidencia una convergencia adecuada y un rol informativo de esta variable en la volatilidad. En contraste, con la covariable PC1 las cadenas presentan mayor tendencia y menor estabilidad, reflejando dificultades de convergencia y estimaciones menos confiables.

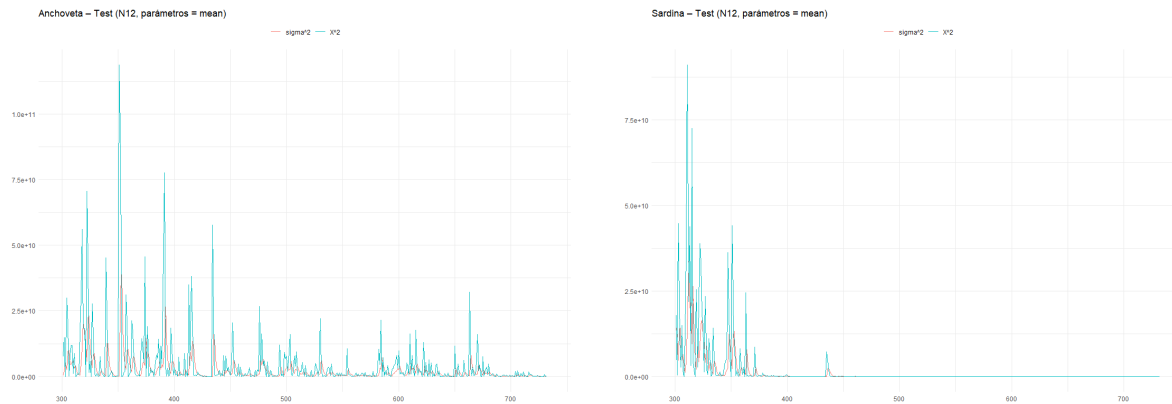


(a) *Traceplot* por parámetro para estimación para la variable Exógena N12.

(b) *Traceplot* por parámetro para estimación para la variable Exógena PC1.

Figura 4.16: Cadenas de las estimaciones para la Sardina.

En la Figura 4.16 se aprecia que, con la covariable N12, las cadenas presentan cierta oscilación pero tienden a concentrarse en rangos definidos, lo que indica una convergencia aceptable aunque con algo de variabilidad. En cambio, con la covariable PC1 las cadenas muestran mayor dispersión y episodios de tendencia, reflejando una convergencia menos clara y estimaciones más inestables.



(a) Evolución de la varianza condicional estimada (σ^2) y los retornos al cuadrado (X^2) para la serie de desembarques de Anchoveta.

(b) Evolución de la varianza condicional estimada (σ^2) y los retornos al cuadrado (X^2) para la serie de desembarques de Sardina.

Figura 4.17: Comparación entre la varianza condicional estimada y los retornos al cuadrado.

En la serie de sardina, los retornos al cuadrado (X^2) muestran episodios de alta volatilidad concentrados principalmente entre los periodos 280 y 370, con picos bien definidos que son seguidos de manera consistente por la varianza condicional estimada (σ^2). El modelo GARCH-X con la covariable N12 logra reproducir adecuadamente la dinámica de agrupamiento de choques, aunque tiende a suavizar la magnitud de algunos valores extremos. A partir del periodo 400, tanto X^2 como σ^2 descienden bruscamente y se estabilizan cerca de cero, lo que refleja un periodo prolongado de calma en la serie.

En el caso de la anchoveta, el comportamiento es más persistente y extendido en el tiempo. Entre los periodos 280 y 450 se observa una sucesión prolongada de picos de gran magnitud, algunos de los cuales superan los $1,0 \times 10^{11}$. La varianza condicional estimada acompaña esta dinámica, mostrando capacidad de capturar la persistencia de los choques y el agrupamiento de volatilidad. A diferencia de la sardina, aquí la inestabilidad se prolonga más allá del periodo 500, con episodios intermitentes que evidencian una estructura más compleja y variable. En conjunto, mientras que en la sardina el modelo reproduce un ciclo de alta volatilidad seguido de calma, en la anchoveta se revela un patrón más prolongado y errático, lo que sugiere mayor sensibilidad a las fluctuaciones ambientales en esta especie.

4.5. Discusión y conclusiones

En la Aproximación Bayesiana Computacional (ABC) la relación entre el tamaño de la tolerancia ε y la dimensión del estadístico de resumen S constituye un compromiso metodológico central. Cuando $\varepsilon \rightarrow 0$, la distribución aproximada converge hacia la posterior verdadera, es decir:

$$\lim_{\varepsilon \rightarrow 0} \pi_{ABC}(\theta | y^{obs}) = \pi(\theta | y^{obs}),$$

lo que garantiza una inferencia más precisa. Sin embargo, este escenario reduce drásticamente la tasa de aceptación, incrementando el costo computacional de manera significativa. En contraste, valores grandes de ε elevan la eficiencia al aceptar un mayor número de simulaciones, pero a costa de introducir sesgo en la aproximación, alejándola de la distribución verdadera e incluso acercándola a la prior en casos extremos.

Para mitigar esta disyuntiva, se han desarrollado esquemas de ε adaptativo que ajustan dinámicamente el umbral en función de la distribución de distancias acumuladas. En este enfoque, ε_t se reduce de manera controlada a lo largo de las iteraciones, comenzando con valores relativamente altos para asegurar una fase inicial de exploración, y descendiendo progresivamente hacia umbrales más estrictos que concentran la posterior alrededor de los valores verdaderos. Este procedimiento permite mantener tasas de aceptación razonables en las primeras etapas y mejorar la precisión en las fases finales, equilibrando así eficiencia y rigor inferencial.

La elección del estadístico de resumen S añade otra dimensión al *trade-off*. Resúmenes de baja dimensión permiten cálculos rápidos y mejoran la tasa de aceptación, pero pueden omitir información crítica de los datos observados, limitando la capacidad de discriminación entre distintos parámetros. Por otro lado, estadísticos de mayor dimensión o más informativos tienden a mejorar el ajuste y aproximar de manera más fiel la posterior verdadera, aunque ello conlleva un incremento sustancial del costo computacional y posibles problemas asociados a una alta dimensionalidad.

En este contexto, la incorporación de resúmenes neuronales, como los obtenidos mediante arquitecturas *TCN+LSTM*, ofrece una alternativa poderosa. Estos modelos son capaces de condensar trayectorias completas en representaciones de baja dimensión que preservan relaciones temporales y no lineales, actuando como estadísticos de resumen más informativos que los basados en momentos simples o transformaciones lineales. Aunque su entrenamiento inicial requiere un esfuerzo computacional considerable, una vez calibrados permiten acelerar el muestreo ABC y mejorar la concentración de las distribuciones posteriores, reduciendo el sesgo sin sacrificar excesivamente la eficiencia.

En consecuencia, el desempeño de los algoritmos ABC depende de alcanzar un equilibrio adecuado: un ε adaptativo que permita transitar desde fases de exploración a fases de concentración, y un conjunto de resúmenes S que combine informatividad y manejabilidad computacional. Este balance constituye uno de los aspectos más críticos para garantizar tanto la validez estadística como la viabilidad práctica de la inferencia en contextos de alta complejidad.

El desarrollo de herramientas sofisticadas para la estimación y predicción en modelos como el GARCH-X no es únicamente un aporte metodológico, sino que constituye una necesidad estratégica en contextos geopolíticos donde la explotación de recursos naturales, como la sardina (*Sardinops sagax*) y la anchoveta (*Engraulis ringens*), juega un rol central en la seguridad alimentaria, el comercio internacional y la soberanía económica de los países costeros. Estas especies forman parte de una de las pesquerías pelágicas más productivas del planeta, pero su dinámica poblacional está fuertemente condicionada por fenómenos ambientales multiescalares (ENSO, cambios de régimen, variabilidad multidecadal), lo que las hace altamente vulnerables a la incertidumbre climática y a presiones antrópicas.

En este marco, disponer de esquemas de inferencia robustos, capaces de integrar información ambiental y capturar dinámicas de volatilidad a través de simulaciones y resúmenes neuronales, permite reducir la incertidumbre en la estimación de parámetros clave y mejorar la capacidad de anticipar escenarios de variabilidad extrema. Esto no solo fortalece la gestión sostenible de los recursos, sino que también otorga herramientas técnicas que respaldan

CAPÍTULO 4. APLICACIÓN A DATOS PESQUEROS DEL NORTE DE CHILE

decisiones políticas y acuerdos internacionales en torno al acceso, explotación y conservación de estos recursos estratégicos. En consecuencia, la implementación de métodos avanzados de regresión y estimación contribuye a equilibrar intereses económicos y ecológicos en un escenario global marcado por tensiones en torno a la explotación de bienes comunes marinos y el desafío de cuidar y hacer uso de los recursos de manera responsable y ética.

Referencias bibliográficas

- [1] Francisco J. Plaza Vega. *Parameter estimation for GARCH-type models with exogenous variables: missing values and machine learning methods*. Doctor en estadística, Instituto de Estadística, Universidad de Valparaíso, Valparaíso, Chile, April 2023. Advisor: Prof. Héctor Araya Carvajal.
- [2] Aston Zhang, Zachary C Lipton, Mu Li, and Alexander J Smola. Dive into deep learning. *arXiv preprint arXiv:2106.11342*, 2021.
- [3] Forecastegy. Multiple time series forecasting with convolutional neural networks in python. <https://forecastegy.com/posts/multiple-time-series-forecasting-with-convolutional-neural-networks-in-python/>, 2023.
- [4] Robert F Engle. Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation. *Econometrica: Journal of the econometric society*, pages 987–1007, 1982.
- [5] Tim Bollerslev. A conditionally heteroskedastic time series model for speculative prices and rates of return. *The review of economics and statistics*, pages 542–547, 1987.
- [6] Bai Jiang, Tung-yu Wu, Charles Zheng, and Wing H Wong. Learning summary statistic for approximate bayesian computation via deep neural network. *Statistica Sinica*, pages 1595–1618, 2017.
- [7] Francisco Plaza-Vega and Héctor Araya. Anchovy (*engraulis ringens*) and pacific sardine (*sardinops sagax*) variability changes in northern chile associated with the environment and inter species synchronicity: Garch model with exogenous variable and hybrid bayesian deep learning estimation approach. *Progress in Oceanography*, 221:103190, 2024.
- [8] Servicio Nacional de Pesca y Acuicultura (SERNAPESCA). Anuarios Estadísticos de Pesca y Acuicultura, 2024.

REFERENCIAS BIBLIOGRÁFICAS

- URL <https://www.sernapesca.cl/informacion-utilidad/anuarios-estadisticos-de-pesca-y-acuicultura/>.
- [9] Dirección Meteorológica de la Armada de Chile (DIRECTEMAR). Sitio web de la Dirección Meteorológica de la Armada de Chile, 2024. URL https://meteoarmada.directemar.cl/prontus_meteo/site/edic/base/port/inicio.html. Accedido el 13 de diciembre de 2024.
- [10] National Oceanic and Atmospheric Administration (NOAA). National Oceanic and Atmospheric Administration Official Website, 2024. URL <https://www.noaa.gov/>. Accedido el 13 de diciembre de 2024.
- [11] Wilfredo Palma. *Time series analysis*. John Wiley & Sons, 2016.
- [12] Peter J Brockwell and Richard A Davis. *Introduction to time series and forecasting*. Springer, 2009.
- [13] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- [14] Wilfredo Palma. *Long-memory time series: theory and methods*. John Wiley & Sons, 2007.
- [15] Robert H Shumway and David S Stoffer. *Time series analysis and its applications: with R examples*. Springer, 2006.
- [16] Christian Francq and Jean-Michel Zakoian. *GARCH models: structure, statistical inference and financial applications*. John Wiley & Sons, 2019.
- [17] Cristiana Tudor. A liquidity-weighted garch model for empirical equity series. In *Proceedings of WSEAS/NAUN International Conferences. Corfu*, 2011.
- [18] Ciprian A Tudor* and Cristiana Tudor. Egarch model with weighted liquidity. *Communications in Statistics-Simulation and Computation*, 43(5):1133–1142, 2014.
- [19] Andreas Fuest and Stefan Mittnik. Modeling liquidity impact on volatility: a garch-funxl approach. *Available at SSRN 3038947*, 2015.
- [20] Genaro Sucarrat, Steffen Grønneberg, and Alvaro Escribano. Estimation and inference in univariate and multivariate log-garch-x models when the conditional density is unknown. *Computational statistics & data analysis*, 100:582–594, 2016.

REFERENCIAS BIBLIOGRÁFICAS

- [21] Pauliina Ilmonen, Soledad Torres, Ciprian Tudor, Lauri Viitasaari, and Marko Voutilainen. On generalized arch model with stationary liquidity. *arXiv preprint arXiv:1806.08608*, 2018.
- [22] Scott A Sisson, Yanan Fan, and Mark Beaumont. *Handbook of approximate Bayesian computation*. CRC press, 2018.
- [23] Paul Marjoram, John Molitor, Vincent Plagnol, and Simon Tavaré. Markov chain monte carlo without likelihoods. *Proceedings of the National Academy of Sciences*, 100(26): 15324–15328, 2003.
- [24] AN Kolmogorov. Definition of center of dispersion and measure of accuracy from a finite number of observations (in russian). *Izv. Akad. Nauk SSSR Ser. Mat.*, 6:3–32, 1942.
- [25] Edward W Barankin and Ashok P Maitra. Generalization of the fisher-darmois-koopman-pitman theorem on sufficient statistics. *Sankhyā: The Indian Journal of Statistics, Series A*, pages 217–244, 1963.
- [26] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [27] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.
- [28] Colin Lea, Michael D. Flynn, Rene Vidal, Austin Reiter, and Gregory D. Hager. Temporal convolutional networks for action segmentation and detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 156–165, 2016. 10.1109/CVPR.2017.113. URL <https://arxiv.org/abs/1611.05267>.
- [29] Shaojie Bai, J. Zico Kolter, and Vladlen Koltun. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. *arXiv preprint arXiv:1803.01271*, 2018. URL <https://arxiv.org/abs/1803.01271>.
- [30] Dennis Prangle. Adapting the abc distance function. 2017.
- [31] Mark A Beaumont, Wenyang Zhang, and David J Balding. Approximate bayesian computation in population genetics. *Genetics*, 162(4):2025–2035, 2002.
- [32] David Ruppert. Efficient estimations from a slowly convergent robbins-monro process. Technical report, Cornell University Operations Research and Industrial Engineering, 1988.

REFERENCIAS BIBLIOGRÁFICAS

- [33] Jorge Valdés, Luc Ortlieb, Dimitri Gutierrez, Luis Marinovic, Gabriel Vargas, and Abdel Sifeddine. 250 years of sardine and anchovy scale deposition record in mejillones bay, northern chile. *Progress in Oceanography*, 79(2-4):198–207, 2008.
- [34] Eleuterio Yáñez, Samuel Hormazábal, Claudio Silva, Aldo Montecinos, M Angela Barbieri, Alexander Valdenegro, Alejandra Órdenes, and Fabián Gómez. Coupling between the environment and the pelagic resources exploited off northern chile: ecosystem indicators and a conceptual model. *Latin American Journal of Aquatic Research*, 36(2):159–181, 2008.
- [35] Instituto Nacional de Estadísticas de Chile. Variables básicas ambientales, 2025. URL <https://www.ine.gob.cl/estadisticas/economia/energia-y-medioambiente/variables-basicas-ambientales>. Accedido el 13 de mayo de 2025.
- [36] Eleuterio Yáñez, Francisco Plaza, Juan Carlos Gutiérrez-Estrada, Nivaldo Rodríguez, MA Barbieri, Inmaculada Pulido-Calvo, and Cinthya Bórquez. Anchovy (*engraulis ringens*) and sardine (*sardinops sagax*) abundance forecast off northern chile: a multivariate ecosystemic neural network approach. *Progress in Oceanography*, 87(1-4): 242–250, 2010.

ANEXOS A

Anexo 1: Estimación híbrida de parámetros con Path Signatures

A.1. Red Neuronal Convolutacional N°1

A.1.1. Descripción de la Arquitectura y Entrenamiento

CNN 1d		
Model: "sequential"		
Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 300, 1024)	7168
max_pooling1d (MaxPooling1D)	(None, 100, 1024)	0
conv1d_1 (Conv1D)	(None, 100, 1024)	4195328
max_pooling1d_1 (MaxPooling1D)	(None, 50, 1024)	0
conv1d_2 (Conv1D)	(None, 50, 1024)	12583936
max_pooling1d_2 (MaxPooling1D)	(None, 25, 1024)	0
conv1d_3 (Conv1D)	(None, 25, 1024)	12583936

ANEXOS A. ANEXO 1: ESTIMACIÓN HÍBRIDA DE PARÁMETROS CON PATH SIGNATURES

max_pooling1d_3 (MaxPoolin g1D)	(None, 12, 1024)	0
conv1d_4 (Conv1D)	(None, 12, 1024)	6292480
max_pooling1d_4 (MaxPoolin g1D)	(None, 6, 1024)	0
conv1d_5 (Conv1D)	(None, 6, 1024)	6292480
max_pooling1d_5 (MaxPoolin g1D)	(None, 3, 1024)	0
conv1d_6 (Conv1D)	(None, 3, 1024)	6292480
flatten (Flatten)	(None, 3072)	0
dropout (Dropout)	(None, 3072)	0
dense (Dense)	(None, 128)	393344
dense_1 (Dense)	(None, 256)	33024
dense_2 (Dense)	(None, 256)	65792
dense_3 (Dense)	(None, 256)	65792
dense_4 (Dense)	(None, 256)	65792
dense_5 (Dense)	(None, 256)	65792
dense_6 (Dense)	(None, 256)	65792
dropout_1 (Dropout)	(None, 256)	0
dense_7 (Dense)	(None, 4)	1028

=====

Total params: 49004164 (186.94 MB)

Trainable params: 49004164 (186.94 MB)

Non-trainable params: 0 (0.00 Byte)

A.1.2. Evolución de Learning Rate

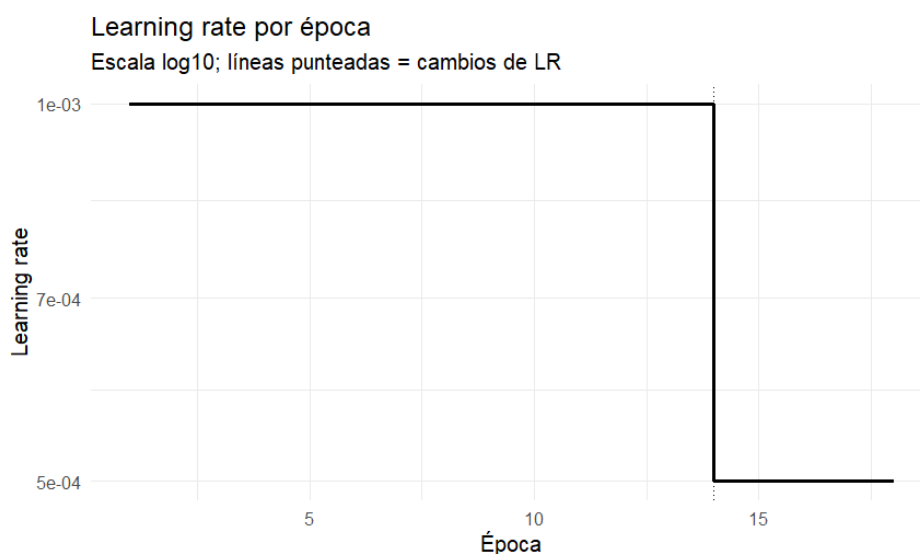


Figura A.1: Evolución de la tasa de aprendizaje durante el entrenamiento de la CNN 1D.

A.2. Red Neuronal Convolutacional + LSTM

A.2.1. Descripción de la Arquitectura y Entrenamiento

CNN 1d

Model: "sequential"

Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 300, 128)	1280
batch_normalization (Batch Normalization)	(None, 300, 128)	512

ANEXOS A. ANEXO 1: ESTIMACIÓN HÍBRIDA DE PARÁMETROS CON PATH SIGNATURES

max_pooling1d (MaxPooling1D)	(None, 150, 128)	0
conv1d_1 (Conv1D)	(None, 150, 256)	98560
batch_normalization_1 (BatchNormalization)	(None, 150, 256)	1024
max_pooling1d_1 (MaxPooling1D)	(None, 75, 256)	0
conv1d_2 (Conv1D)	(None, 75, 256)	327936
batch_normalization_2 (BatchNormalization)	(None, 75, 256)	1024
max_pooling1d_2 (MaxPooling1D)	(None, 37, 256)	0
conv1d_3 (Conv1D)	(None, 37, 512)	655872
batch_normalization_3 (BatchNormalization)	(None, 37, 512)	2048
max_pooling1d_3 (MaxPooling1D)	(None, 18, 512)	0
lstm (LSTM)	(None, 18, 128)	328192
dropout (Dropout)	(None, 18, 128)	0
lstm_1 (LSTM)	(None, 128)	131584
dropout_1 (Dropout)	(None, 128)	0
dense (Dense)	(None, 256)	33024
batch_normalization_4 (BatchNormalization)	(None, 256)	1024

ANEXOS A. ANEXO 1: ESTIMACIÓN HÍBRIDA DE PARÁMETROS CON PATH SIGNATURES

chNormalization)		
dropout_2 (Dropout)	(None, 256)	0
dense_1 (Dense)	(None, 128)	32896
batch_normalization_5 (Bat chNormalization)	(None, 128)	512
dropout_3 (Dropout)	(None, 128)	0
dense_2 (Dense)	(None, 64)	8256
batch_normalization_6 (Bat chNormalization)	(None, 64)	256
dropout_4 (Dropout)	(None, 64)	0
dense_3 (Dense)	(None, 4)	260
=====		
Total params: 1624260 (6.20 MB)		
Trainable params: 1621060 (6.18 MB)		
Non-trainable params: 3200 (12.50 KB)		

A.2.2. Evolución de Learning Rate

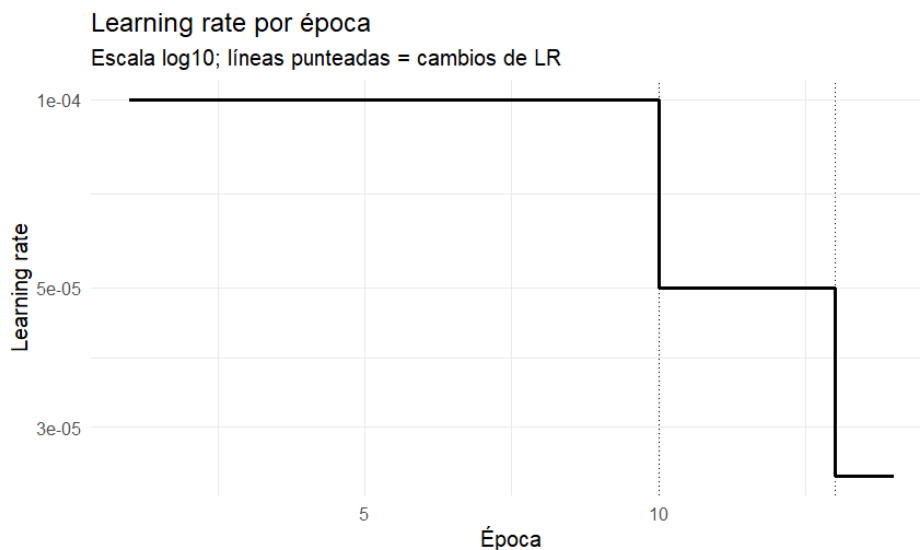


Figura A.2: Evolución de la tasa de aprendizaje durante el entrenamiento de la CNN + LSTM.

A.3. Red Neuronal Convolutiva N°3

A.3.1. Descripción de la Arquitectura y Entrenamiento

CNN 1d		
Model: "sequential"		
Layer (type)	Output Shape	Param #
conv1d (Conv1D)	(None, 300, 1024)	7168
max_pooling1d (MaxPooling1D)	(None, 100, 1024)	0
conv1d_1 (Conv1D)	(None, 100, 1024)	4195328
max_pooling1d_1 (MaxPooling1D)	(None, 50, 1024)	0
conv1d_2 (Conv1D)	(None, 50, 1024)	12583936

ANEXOS A. ANEXO 1: ESTIMACIÓN HÍBRIDA DE PARÁMETROS CON PATH SIGNATURES

max_pooling1d_2 (MaxPooling1D)	(None, 25, 1024)	0
conv1d_3 (Conv1D)	(None, 25, 1024)	12583936
max_pooling1d_3 (MaxPooling1D)	(None, 12, 1024)	0
conv1d_4 (Conv1D)	(None, 12, 1024)	6292480
max_pooling1d_4 (MaxPooling1D)	(None, 6, 1024)	0
conv1d_5 (Conv1D)	(None, 6, 1024)	6292480
max_pooling1d_5 (MaxPooling1D)	(None, 3, 1024)	0
conv1d_6 (Conv1D)	(None, 3, 1024)	6292480
flatten (Flatten)	(None, 3072)	0
dropout (Dropout)	(None, 3072)	0
dense (Dense)	(None, 128)	393344
dense_1 (Dense)	(None, 256)	33024
dense_2 (Dense)	(None, 256)	65792
dense_3 (Dense)	(None, 256)	65792
dense_4 (Dense)	(None, 256)	65792
dense_5 (Dense)	(None, 256)	65792
dense_6 (Dense)	(None, 256)	65792
dropout_1 (Dropout)	(None, 256)	0

dense_7 (Dense)	(None, 4)	1028
=====		
Total params: 49004164 (186.94 MB)		
Trainable params: 49004164 (186.94 MB)		
Non-trainable params: 0 (0.00 Byte)		

A.3.2. Evolución de Learning Rate

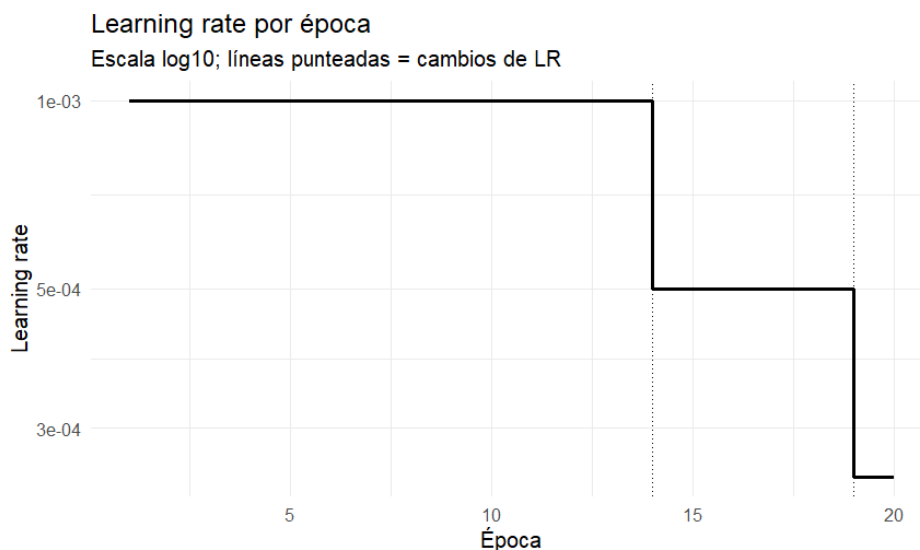


Figura A.3: Evolución de la tasa de aprendizaje durante el entrenamiento de la CNN 1D N°3.

A.4. Red neuronal TCN + LSTM N°5

A.4.1. Descripción de la Arquitectura y Entrenamiento

TCN + LSTM		
Model: "CNN_LSTM_TCN_expanded"		

Layer (type)	Output Shape	Param #
Connected to		

ANEXOS A. ANEXO 1: ESTIMACIÓN HÍBRIDA DE PARÁMETROS CON PATH SIGNATURES

input_series (InputLayer)	[(None, 300, 3)]	0	[
conv1d (Conv1D)	(None, 300, 512)	5120	['
input_series[0][0]'			
batch_normalization (Batch Normalization)	(None, 300, 512)	2048	['
conv1d[0][0]'			
spatial_dropout1d (Spatial Dropout1D)	(None, 300, 512)	0	['
batch_normalization[0][0]'			
conv1d_1 (Conv1D)	(None, 300, 512)	786944	['
spatial_dropout1d[0][0]'			
conv1d_2 (Conv1D)	(None, 300, 512)	2048	['
input_series[0][0]'			
batch_normalization_1 (Batch Normalization)	(None, 300, 512)	2048	['
conv1d_1[0][0]'			
add (Add)	(None, 300, 512)	0	['
conv1d_2[0][0]'			,
			batch_normalization_1[0][0]'
]
conv1d_3 (Conv1D)	(None, 300, 512)	786944	['
add[0][0]'			
batch_normalization_2 (Batch Normalization)	(None, 300, 512)	2048	['
conv1d_3[0][0]'			
spatial_dropout1d_1 (Spatial Dropout1D)	(None, 300, 512)	0	['
batch_normalization_2[0][0]']
conv1d_4 (Conv1D)	(None, 300, 512)	786944	['
spatial_dropout1d_1[0][0]'			
batch_normalization_3 (Batch Normalization)	(None, 300, 512)	2048	['

ANEXOS A. ANEXO 1: ESTIMACIÓN HÍBRIDA DE PARÁMETROS CON PATH SIGNATURES

conv1d_4[0][0]'] chNormalization)			
add_1 (Add) add[0][0]',	(None, 300, 512)	0	[' , batch_normaliz [0][0]'
conv1d_5 (Conv1D) add_1[0][0]']	(None, 300, 512)	786944	['
batch_normalization_4 (Bat conv1d_5[0][0]'] chNormalization)	(None, 300, 512)	2048	['
spatial_dropout1d_2 (Spati batch_normalization_4[0][0]' alDropout1D)	(None, 300, 512)	0	[']
conv1d_6 (Conv1D) spatial_dropout1d_2[0][0]']	(None, 300, 512)	786944	['
batch_normalization_5 (Bat conv1d_6[0][0]'] chNormalization)	(None, 300, 512)	2048	['
add_2 (Add) add_1[0][0]',	(None, 300, 512)	0	[' , batch_normaliz [0][0]'
conv1d_7 (Conv1D) add_2[0][0]']	(None, 300, 512)	786944	['
batch_normalization_6 (Bat conv1d_7[0][0]'] chNormalization)	(None, 300, 512)	2048	['
spatial_dropout1d_3 (Spati batch_normalization_6[0][0]' alDropout1D)	(None, 300, 512)	0	[']

ANEXOS A. ANEXO 1: ESTIMACIÓN HÍBRIDA DE PARÁMETROS CON PATH SIGNATURES

conv1d_8 (Conv1D) spatial_dropout1d_3[0][0]'	(None, 300, 512)	786944	['
batch_normalization_7 (Batch Normalization) conv1d_8[0][0]'	(None, 300, 512)	2048	['
add_3 (Add) add_2[0][0]',	(None, 300, 512)	0	['
			,
]
conv1d_9 (Conv1D) add_3[0][0]'	(None, 300, 512)	786944	['
batch_normalization_8 (Batch Normalization) conv1d_9[0][0]'	(None, 300, 512)	2048	['
spatial_dropout1d_4 (Spatial Dropout) batch_normalization_8[0][0]'	(None, 300, 512)	0	['
]
conv1d_10 (Conv1D) spatial_dropout1d_4[0][0]'	(None, 300, 512)	786944	['
batch_normalization_9 (Batch Normalization) conv1d_10[0][0]'	(None, 300, 512)	2048	['
add_4 (Add) add_3[0][0]',	(None, 300, 512)	0	['
			,
]
conv1d_11 (Conv1D) add_4[0][0]'	(None, 300, 512)	786944	['
batch_normalization_10 (Batch Normalization) conv1d_11[0][0]'	(None, 300, 512)	2048	['

batch_normaliz
[0][0]'

batch_normaliz
[0][0]'

ANEXOS A. ANEXO 1: ESTIMACIÓN HÍBRIDA DE PARÁMETROS CON PATH SIGNATURES

spatial_dropout1d_5 (SpatialDropout1D)	(None, 300, 512)	0	['batch_normalization_10[0][0]
conv1d_12 (Conv1D)	(None, 300, 512)	786944	spatial_dropout1d_5[0][0]']
batch_normalization_11 (BatchNormalization)	(None, 300, 512)	2048	conv1d_12[0][0]']
add_5 (Add)	(None, 300, 512)	0	batch_normalization_11[0][0]
add_4[0][0]',			']
lstm (LSTM)	(None, 300, 512)	2099200	add_5[0][0]']
dropout (Dropout)	(None, 300, 512)	0	lstm[0][0]']
lstm_1 (LSTM)	(None, 300, 512)	2099200	dropout[0][0]']
dropout_1 (Dropout)	(None, 300, 512)	0	lstm_1[0][0]']
dense (Dense)	(None, 300, 1)	513	dropout_1[0][0]']
softmax (Softmax)	(None, 300, 1)	0	dense[0][0]']
multiply (Multiply)	(None, 300, 512)	0	softmax[0][0]']
multiply[0][0]',			']
lambda (Lambda)	(None, 512)	0	multiply[0][0]']
dense_1 (Dense)	(None, 512)	262656	lambda[0][0]']

ANEXOS A. ANEXO 1: ESTIMACIÓN HÍBRIDA DE PARÁMETROS CON PATH SIGNATURES

lambda[0][0]'			
batch_normalization_12 (Batch Normalization)	(None, 512)	2048	['
dense_1[0][0]'			
tchNormalization)			
dropout_2 (Dropout)	(None, 512)	0	['
batch_normalization_12[0][0]			']
dense_2 (Dense)	(None, 256)	131328	['
dropout_2[0][0]'			
batch_normalization_13 (Batch Normalization)	(None, 256)	1024	['
dense_2[0][0]'			
tchNormalization)			
dropout_3 (Dropout)	(None, 256)	0	['
batch_normalization_13[0][0]			']
dense_3 (Dense)	(None, 128)	32896	['
dropout_3[0][0]'			
batch_normalization_14 (Batch Normalization)	(None, 128)	512	['
dense_3[0][0]'			
tchNormalization)			
dropout_4 (Dropout)	(None, 128)	0	['
batch_normalization_14[0][0]			']
dense_4 (Dense)	(None, 64)	8256	['
dropout_4[0][0]'			
batch_normalization_15 (Batch Normalization)	(None, 64)	256	['
dense_4[0][0]'			
tchNormalization)			
dropout_5 (Dropout)	(None, 64)	0	['
batch_normalization_15[0][0]			']
params (Dense)	(None, 4)	260	['
dropout_5[0][0]'			
=====			

Total params: 13326277 (50.84 MB)
Trainable params: 13312069 (50.78 MB)
Non-trainable params: 14208 (55.50 KB)

A.4.2. Evolución de Learning Rate

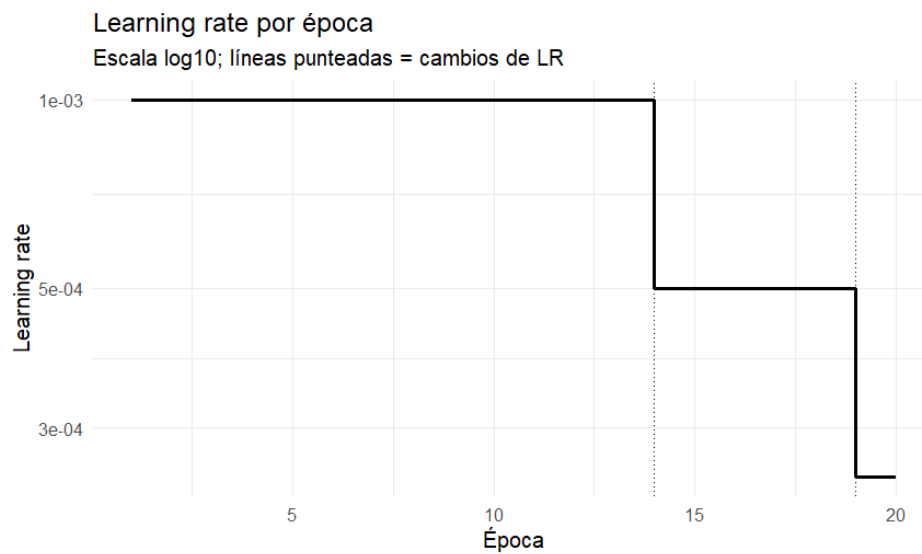


Figura A.4: Evolución de la tasa de aprendizaje durante el entrenamiento de la TCN + LSTM.

ANEXOS B

Anexo 2: Análisis exploratorio de datos

B.1. Visualización cuantil cuantil de variables ambientales

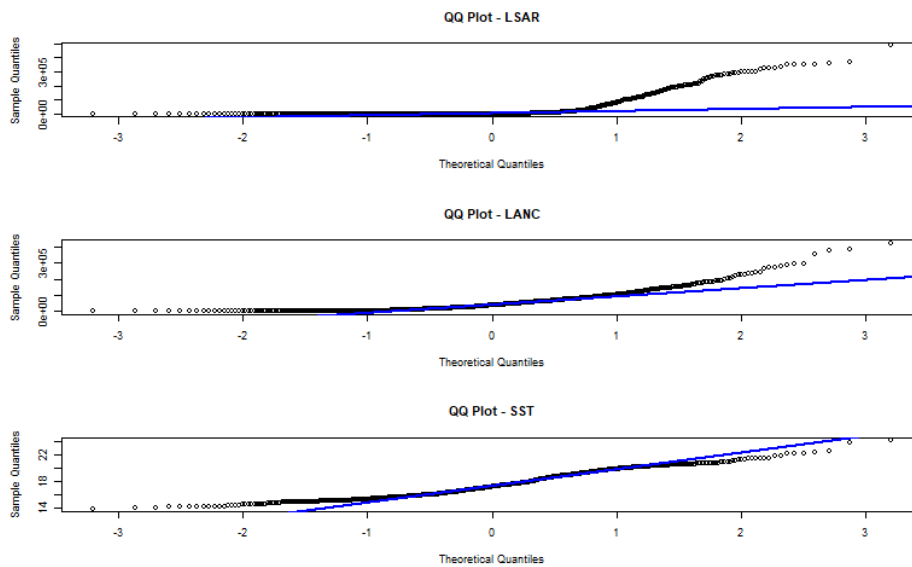


Figura B.1: Gráfico cuantil-cuantil de las variables ambientales *LSAR*, *LANC*, *SST*.

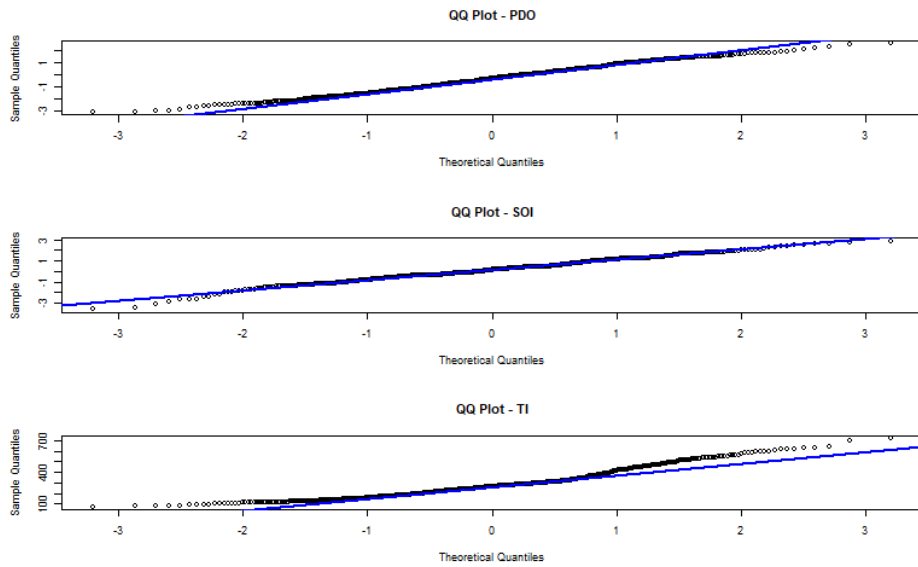


Figura B.2: Gráfico cuantil-cuantil de las variables ambientales *PDO*, *SOI*, *TI*.

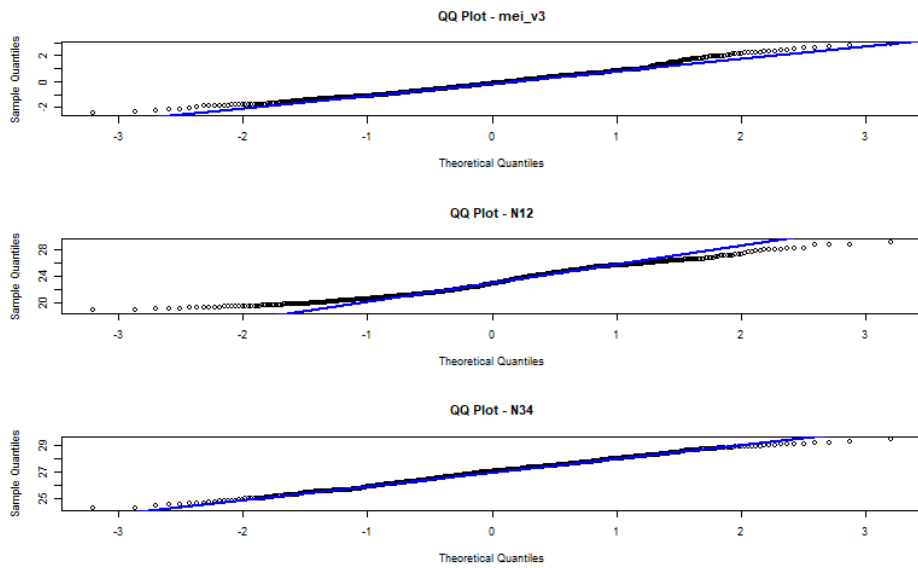


Figura B.3: Gráfico cuantil-cuantil de las variables ambientales *MEI*, *N12*, *N34*.

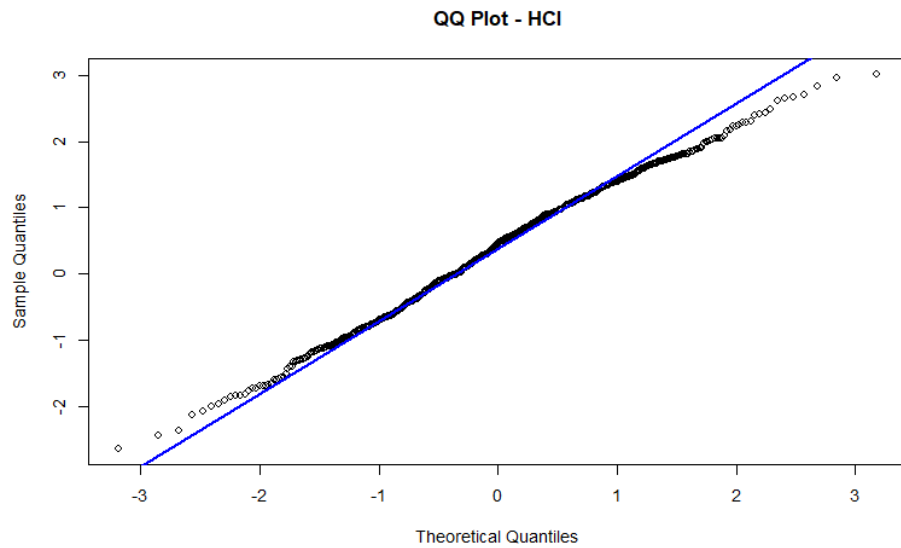


Figura B.4: Gráfico cuantil-cuantil de la variable *HCI*.

B.2. Visualización cuantil cuantil de retonos de desembarque de Sardina y Anchoveta

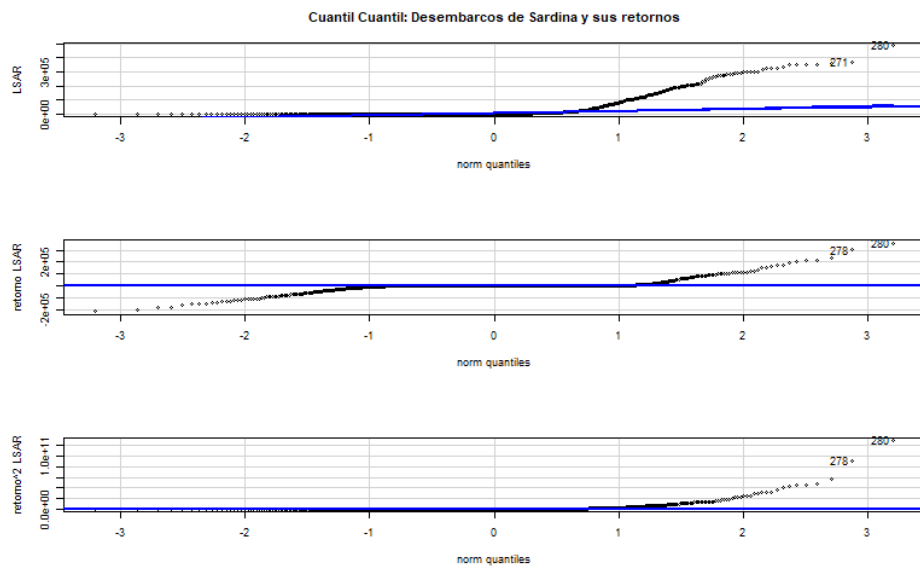


Figura B.5: Gráfico cuantil-cuantil de retornos LSAR.

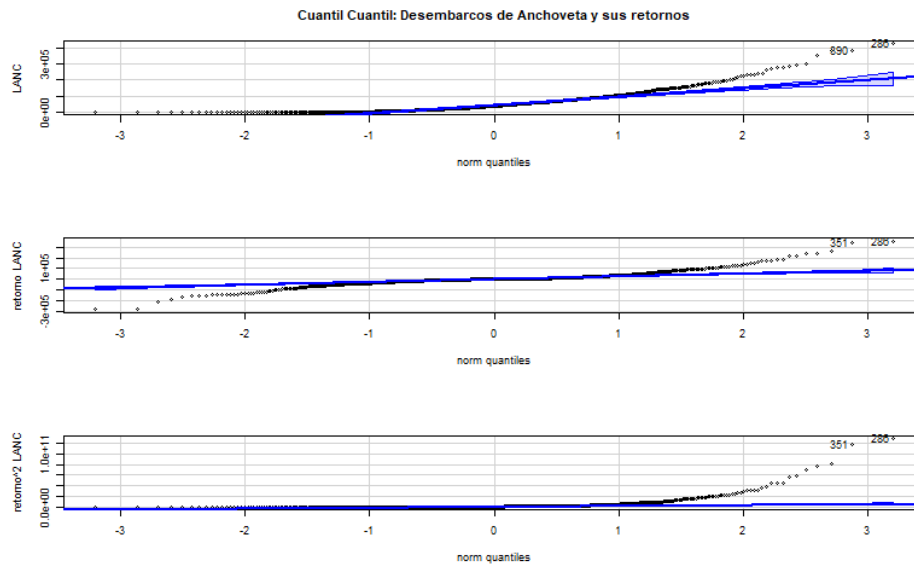


Figura B.6: Gráfico cuantil-cuantil de retornos LANC.

B.3. Codigos del Proyecto

Los códigos del proyecto están disponibles en el siguiente repositorio de GitHub:
<https://github.com/Bastian-Rafael/CODIGOS-TESIS-A2>