



Universidad del Bío-Bío, Chile
Facultad de Ciencias Empresariales

DEEP LEARNING PARA LA ESTIMACIÓN DE PRECIOS
EN LA INDUSTRIA DEL SALMÓN EN CHILE

TESIS PRESENTADA POR ROBERTO VILLARROEL VELÁSQUEZ PARA
OBTENER EL GRADO DE MAGÍSTER EN CIENCIAS DE LA COMPUTACIÓN

DIRIGIDA POR

DR. PEDRO CAMPOS SOTO

2022

Agradecimientos

A mi familia por el apoyo incondicional y mis profesores Pedro Campos y César Salazar, por el tiempo, dedicación y conocimiento entregado durante este período.

El desarrollo de esta Tesis fue apoyado por la Universidad del Bío-Bío, a través de los proyectos 230248 IF/R y 194810 GI/VC

Resumen

La estimación de precios en distintas áreas es uno de los desafíos de mayor importancia en la actualidad. Muchos investigadores han estudiado como obtener la tendencia del mercado y poder estimar el precio en distintos sectores económicos y productos. En efecto, numerosos y variados modelos han sido propuestos para este tipo de tarea, centrados principalmente en modelos estadísticos y modelos tradicionales de Machine Learning como Support Vector Machines y Redes Neuronales clásicas. Por otro lado, la salmonicultura en Chile es uno de los sectores de mayor importancia a nivel de comercio exterior, por tanto, contar con modelos predictivos en función de datos públicos resulta de gran relevancia para empresas en esta industria, y además, podría establecer parámetros iniciales para el desarrollo de políticas públicas que impulsen el crecimiento del sector en Chile. Por lo anterior, en este trabajo se propone evaluar distintos modelos de *Deep Learning* (DL) para la estimación de precios de salmón en Chile, generando información útil para la toma de decisiones y al mismo tiempo, comparar su rendimiento frente a modelos tradicionales utilizados para este tipo de problemas. Para lo anterior, se emplean datos del Servicio Nacional de Aduanas, Banco Central, Banco Mundial, Eurostats y la Oficina de Estadísticas Laborales de Estados Unidos. Con estos datos se entrenan modelos de DL como *Long-Short Term Memory* (LSTM) y *Convolutional Neural Networks* (CNN), los que son contrastados con modelos ARIMA, y modelos clásicos de *Machine Learning*. Se contempla la optimización de hiperparámetros, midiendo su desempeño mediante métricas bien establecidas para la evaluación de estos modelos, tales como MAE, MSE y R^2 . En total se realizaron 117 experimentos, utilizando seis series temporales con diferentes características.

Finalmente, en las pruebas finales, las estimaciones obtenidas mediante los modelos LSTM y CNN 1D logran un mejor desempeño en relación al baseline.

Palabras claves - Estimación de precios, *Deep Learning*, *Long-Short Term Memory*, *Convolutional Neural Networks*.

Abstract

Price estimation in different areas is one of the most important challenges today. Many researchers have studied how to obtain the market trend and be able to estimate the price in different economic sectors and products. Indeed, numerous and varied models have been proposed for this type of task, mainly focused on statistical models and traditional Machine Learning models such as Support Vector Machines and classic Neural Networks. On the other hand, salmon farming in Chile is one of the most important sectors in terms of foreign trade, therefore, having predictive models based on public data is of great relevance for companies in this industry, and could also establish initial parameters for the development of public policies that promote the growth of the sector in Chile. Therefore, in this work we propose to evaluate different Deep Learning (DL) models for estimating salmon prices in Chile, generating useful information for decision making and at the same time, comparing their performance against traditional models used for this type of problems. For this purpose, data from the National Customs Service data from the National Customas Service and Central Bank of Chile, as well as, World Bank, Eurostats and the United States Bureau of Labor Statistics. With this data, DL models such as Long-Short Term Memory (LSTM) and Convolutional Neural Networks (CNN) are trained, which are contrasted with ARIMA models and classical Machine Learning models. The optimization of hyperparameters is contemplated, measuring their performance by means of well-established metrics for the evaluation of these models, such as MAE, MSE and R2. In total, 117 experiments were performed, using six time series with different characteristics.

Finally, in the final tests, the estimates obtained using the LSTM and CNN 1D models achieve a better performance relative to the baseline

Keywords - Price estimation, *Deep Learning*, *Long-Short Term Memory*, *Convolutional Neural Networks*.

Índice general

1. Introducción	1
1.1. Propuesta de Tesis	2
1.1.1. Problema	2
1.1.2. Hipótesis	3
1.1.3. Objetivos	3
1.1.4. Alcance de la Investigación	3
1.1.5. Metodología de Trabajo	4
1.2. Organización de informe de Tesis	4
2. Conceptos Preliminares	6
2.1. Estimación de precios del salmón	6
2.1.1. Proceso productivo del salmón	6
2.1.2. Especies salmonídeas	6
2.1.3. Modelos predictivos en la estimación de precios	7
2.2. <i>Machine Learning</i>	9
2.3. <i>Neural Networks</i>	11
2.3.1. Función de activación	13
2.3.2. Optimizadores	13
2.4. <i>Deep Learning</i>	14
2.4.1. <i>Feedforward Neural Networks</i> (FNN)	14
2.4.2. <i>Recurrent Neural Networks</i> (RNN)	15
2.4.3. <i>Long Short - Term Memory</i> (LSTM)	15
2.4.4. <i>Convolutional Neural Networks</i> (CNN)	18
2.4.5. <i>1D Convolutional Neural Networks</i>	19
3. Estado del Arte	20
3.1. Proceso de revisión	20
3.2. Resultados	21
3.2.1. Estimación de precios utilizando modelos de ML y DL en el mercado del salmón	21
3.2.2. Estimación de precios utilizando modelos de ML y DL	21
3.3. Síntesis	24
4. Implementación	25
4.1. Conjuntos de datos	25
4.2. Preprocesamiento de datos	27
4.2.1. Construcción variable <i>price</i>	27
4.2.2. Diseño y elaboración de series temporales	29
4.3. Modelos implementados	32
4.3.1. Pycaret	34
4.3.2. AutoKeras	34
5. Resultados y Discusión	35
5.1. Diseño experimental	35
5.2. Modelo <i>ARIMA</i>	36
5.3. Modelos clásicos de Machine Learning	37
5.4. Modelos Deep Learning con ajuste automático	38
5.5. Long Short-Term Memory	38

5.5.1. Serie Salmón Salar	39
5.5.2. Análisis otras series temporales	45
5.6. Convolutional Neural Networks	47
5.7. Análisis comparativos	47
5.7.1. Sobre optimización de hiperparámetros	48
6. Conclusiones y Trabajos Futuros	50
6.1. Trabajos futuros	52
Anexo 1	53
Anexo 2	58
Anexo 3	62

Índice de figuras

2.1. Ciclo productivo salmón (disponible en:www.salmonchile.cl).	6
2.2. Esquema de Inteligencia Artificial y Sub-Campos (www.colab.research.google.com).	9
2.3. Arquitectura básica de un perceptrón (Aggarwal 2018).	11
2.4. Red Neuronal Artificial (Perry 2018).	12
2.5. Mínimo local y mínimo global. Basado en (Chollet, 2021)	13
2.6. SGD pérdida de superficie. Basado en (Chollet, 2021)	14
2.7. Esquema de <i>Feedforward Neural Networks</i>	15
2.8. Módulo de repetición RNN de una sola capa (https://colah.github.io)	15
2.9. Módulo repetitivo en una red LSTM (https://colah.github.io)	16
2.10. Etapa inicial de cálculo red LSTM	16
2.11. Segunda Etapa de cálculo red LSTM	17
2.12. Tercera Etapa de cálculo red LSTM	17
2.13. Cuarta Etapa de cálculo red LSTM	17
2.14. Arquitectura de CNN's (Ting 2019)	18
2.15. 1D CNN para series de tiempo (Kim 2014).	19
4.1. Conjuntos de datos y su composición	25
4.2. Histograma para la variable fobus	28
4.3. Histograma para la variables peso bruto (kg)	28
4.4. Series de Tiempo Salmón Salar	31
4.5. Diagrama experimentación implementada	32
4.6. Esquemas de capas en implementación de cada modelo	33
5.1. Métricas N° de Neuronas Serie de Tiempo Salmón Salar.	39
5.2. Métricas N° de Capas Serie de Tiempo Salmón Salar.	40
5.3. Métricas Función de Activación Serie de Tiempo Salmón Salar.	41
5.4. Métricas Dropout Recurrente Serie de Tiempo Salmón Salar.	42
5.5. Métricas Dropout Serie de Tiempo Salmón Salar.	43
5.6. Métricas Optimizador Serie de Tiempo Salmón Salar.	44

Índice de cuadros

3.1. Resumen de trabajos relacionados en la estimación de precios en el mercado del salmón	21
3.2. Resumen de trabajos relacionados en la estimación de precios	22
4.1. Variables del conjunto de datos obtenidos desde el Servicio Nacional de Aduanas (SNA)	26
4.2. Conjunto de datos adicionales para el modelamiento	26
4.3. Series de tiempo períodos de inicio y término	30
5.1. Resumen Métricas ARIMA	36
5.2. Principales modelos con mejor rendimiento	37
5.3. Resumen Métricas Autokeras	38
5.4. Resumen Métricas CNN	47
5.5. Principales modelos y resultados	47
6.1. Modelos con resultados más altos de métricas	51
6.2. Resumen Modelos clásicos con mejor rendimiento utilizando PyCaret	53
6.3. Resumen hiperparámetros Serie Salmón Salar	54
6.4. Resumen de métricas función de activación	55
6.5. Resumen de métricas Dropout	56
6.6. Resumen de métricas Optimizador	57
6.7. Partida Arancelaria (302) Pescado fresco o refrigerado, excepto los filetes y demás carne de pescado de la partida 03.04	58
6.8. Partida Arancelaria (303) Pescado congelado, excepto los filetes y demás carne de pescado de la partida 03.04.	59
6.9. Partida Arancelaria (304) Filetes y demás carne de pescado (incluso picada), frescos, refrigerados o congelados	60
6.10. Partida Arancelaria (305) Pescado seco, salado o en salmuera; pescado ahumado, incluso cocido antes o durante el ahumado; harina, polvo y «pellets» de pescado, aptos para la alimentación humana	61
6.11. Partida Arancelaria 302 Salmón Salar Pescado fresco o refrigerado, excepto los filetes y demás carne de pescado de la partida 03.04	62
6.12. Partida Arancelaria 303 Salmón Salar Pescado congelado, excepto los filetes y demás carne de pescado de la partida 03.04.	62
6.13. Partida Arancelaria 304 Salmón Salar Filetes y demás carne de pescado (incluso picada), frescos, refrigerados o congelados.	62
6.14. Partida Arancelaria 305 Salmón Salar Pescado seco, salado o en salmuera; pescado ahumado, incluso cocido antes o durante el ahumado; harina, polvo y «pellets» de pescado, aptos para la alimentación humana.	63

Capítulo 1

Introducción

La acuicultura es una práctica antigua, que se desarrolla a partir de la observación de la conducta de los animales acuáticos, sus necesidades y patrones de comportamiento, permitiendo replicar artificialmente su desarrollo. De esta manera, mediante el cultivo controlado, se garantizan las provisiones alimenticias sin dependencia absoluta de la captura. Para la *Food and Agriculture Organization* (FAO), la acuicultura contempla el cultivo de peces, moluscos, crustáceos y plantas acuáticas, donde la intervención del hombre hace posible la mejora productiva en procesos tales como: la siembra, alimentación y protección frente a depredadores naturales, entre otros aspectos [1].

Durante el año 2018¹ se estima que la producción mundial de pescado alcanzó unos 179 millones de toneladas, siendo un 46,0% generado por la acuicultura. En términos de valorización, para el mismo año, la producción acuícola mundial generó 263.600 millones de dólares. Adicionalmente, la tasa media de crecimiento anual del consumo total de pescado comestible aumentó un 3,1%, superando la tasa de crecimiento anual de la población entre 1961-2017 (1,6%). A nivel mundial, desde 2016, la acuicultura ha sido la principal fuente de pescado disponible para el consumo humano. En 2018, esta proporción era del 52%, una cifra que se puede esperar que siga aumentando a largo plazo. La acuicultura ha ampliado la disponibilidad de pescado a regiones y países que de otro modo tendrían un acceso limitado o nulo a las especies cultivadas, a menudo a precios más bajos, lo que ha conducido a una mejora de la nutrición y la seguridad alimentaria. En valores per cápita, el consumo de pescado comestible aumentó de 9,0 kg (equivalente en peso vivo) en 1961 a 20,3 kg en 2017. Las estimaciones preliminares del consumo de pescado per cápita en 2018 se sitúan actualmente en 20,5 kg [2].

Por tanto, el gran potencial con el que cuenta la Macro Zona Sur de Chile (Biobío, La Araucanía, Los Lagos, Aysén y Magallanes²) en procesos de cultivo de salmón, posicionan al país como el segundo productor de especies salmonídeas a nivel mundial, después de Noruega y al mismo tiempo, esto representó un 14,4% de las exportaciones no cobre durante el año 2019³. Por otro lado, Chile concentró en torno a un 26,0% de la producción⁴ a nivel mundial durante el año 2020, totalizando exportaciones de salmón y trucha por US\$4.382 millones⁵. Por otro lado, en el año 2021 las exportaciones de salmón y trucha totalizaron US\$5.180 millones⁶, demostrando que las exportaciones de salmones y truchas presentan un crecimiento importante durante los últimos 10 años, al mismo tiempo, que los precios ostentan un incremento favorable durante este último tiempo.

Actualmente, los productores de salmón adaptan su programa de producción con el objetivo de maximizar sus beneficios, dependiendo de la trayectoria esperada del *spot price*⁷, basando sus operaciones en expectativas del nivel de costo de entrada y que estará definido por el *spot price* [3]. Por lo tanto, es de suma relevancia establecer modelos predictivos más precisos, que puedan aportar al desarrollo de la salmonicultura en Chile, y poder contar con modelos de estimación de precios para el sector. Actualmente, se utilizan modelos estadísticos y modelos tradicionales como Support Vector Machines, Redes Neuronales clásicas entre otros, para estimar precios de diferentes mercados, incluidos la industria del salmón. Por otro lado, los modelos *Deep Learning* tienen una gran capacidad de aprendizaje y resolución para este

¹El Estado Mundial de la Pesca y la Acuicultura 2020, disponible en www.fao.org.

²Estas cinco regiones componen el alcance geográfico a nivel de Macro Zona Sur.

³La exportación no cobre del año 2019 en Chile totalizaron en valor FOB MMUS\$35.623, según registros del Servicio Nacional de Aduanas (<https://www.aduana.cl>).

⁴Disponible en <https://www.consejodelsalmon.cl/informacion-de-la-industria/estudios-del-consejo/>

⁵Disponible en Banco Central de Chile (<https://www.bcentral.cl/>)

⁶Disponible en Servicio Nacional de Aduanas (<https://www.aduana.cl/>)

⁷Se define como el precio de pago al contado de un bien de acuerdo a los términos del contrato. Serie disponible y publicada de forma semanal en NASDAQ.

tipo de problemas [7], por tanto, es posible establecer modelos predictivos más precisos para la estimación de precios en el sector salmonicultor, mediante el uso de modelos de *Deep Learning*.

Es necesario hacer esfuerzos en desarrollar para potenciar el uso de los datos administrativos para la estadística oficial con el objetivo de complementar, contrastar o simplemente reemplazar otra fuente de menor calidad y cobertura.

En la literatura financiera, se destacan como los métodos más utilizados en la predicción de precios y análisis de series temporales, los modelos ARIMA, SARIMA, VAR, *Machine Learning* (ML) y *Deep Learning* (DL). A través del descubrimiento de patrones en datos históricos, los modelos de ML y DL, pretenden identificar las funciones subyacentes sobre las que se forman los datos y realizar modelos lineales y no lineales. En los últimos años, los modelos como *Neural Networks* (NN) y *Support Vector Machines* (SVM) han sido ampliamente utilizados en la predicción de series financieras, teniendo altos niveles de precisión en predicciones. Resultados de diferentes estudios consideran que estos métodos presentan mejores resultados en la predicción de precios y análisis de series temporales que métodos tradicionales como los expuestos en [4][5][6] y [7]. Por este motivo, en muchos países como Estados Unidos, Alemania, Francia y el Reino Unido, mediante el uso de DL han obtenido mejores predicciones para diferentes productos [7].

En ese sentido, utilizar modelos de series de tiempo basados en DL para la predicción de precios en la industria del salmón, podría permitir generar un instrumento más sólido y con mayor precisión, para el desarrollo del sector, con el fin de planificar y optimizar los recursos, ajustando las demandas establecidas por los mercados externos [8]. Esto a su vez, podría facilitar el desarrollo de políticas públicas que puedan aportar al crecimiento de la industria, y también generar relevancia en el uso de datos públicos tales como Registros Administrativos (RRAA)⁸ para la toma de decisiones.

1.1. Propuesta de Tesis

1.1.1. Problema

El modelamiento de series temporales históricamente ha sido un área de investigación académica, formando una parte integral en diferentes tipos de aplicaciones, por ejemplo, modelamiento del clima, ciencias biológicas, decisiones comerciales, marketing y finanzas, entre otras. Por un lado, los métodos tradicionales se han centrado en modelos paramétricos, tales como modelos autorregresivos (AR), suavizamiento exponencial y modelos de series de tiempo. Por otro lado, los métodos basados en ML ofrecen una dinámica de aprendizaje centrada únicamente en datos. Del mismo modo, los modelos basados en DL han ganado popularidad durante el último tiempo, debido a su capacidad de procesar representaciones complejas de datos, estableciendo modelos más potentes y exactos al momento de generar estimaciones [9].

En ese sentido, la estimación de precios se presenta como un problema centrado en series temporales, que puede ser desarrollado por arquitecturas de DL debido a las distintas posibilidades que ofrece para este tipo de problemas, que involucra determinar el comportamiento de los precios de salmón en diferentes mercados de interés. Para esto se propone el uso de RRAA, con el fin de utilizar información pública disponible que apoye a la toma de decisiones, en función de la demanda generada por múltiples mercados, y al mismo tiempo, planificar recursos y establecer políticas públicas que puedan aportar al desarrollo de la industria salmonera en Chile con herramientas basadas en el análisis de datos.

Una primera aproximación es la integración de datos y RRAA provenientes del Servicio Nacional de Aduanas (SNA), referentes a las exportaciones de productos derivados de salmón y trucha. Dicha información se encuentra en valores FOB⁹, con desagregación a nivel de país de destino, mes de exportación, puerto de destino, cantidad de mercancía exportada y otras variables de interés que puedan aportar al desarrollo de un modelo predictivo a fin de fortalecer las estimaciones.

⁸Registros Administrativos – RRAA, corresponden a información en formato de dataset disponible para elaborar estudios y/o indicadores por distintos entes públicos, privados, academia y público en general.

⁹Valor de una mercancía de exportación cuando está embarcada en la nave, antes de salir hacia su destino ([www.https://old.hacienda.cl/glosario/valor-fob.html](https://old.hacienda.cl/glosario/valor-fob.html)).

En función de lo expuesto anteriormente, en esta tesis se propone el desarrollo de modelos de estimación de $\hat{y}_{i,t+1}$ correspondientes al precio de exportación de salmónidos en valores US/kg, según se describe en la ecuación (1.1):

$$\hat{y}_{i,t+1} = f(y_{i,t-k}, x_{i,t-k:t}, s_i) \quad (1.1)$$

Donde, la estimación $\hat{y}_{i,t+1}$ se calcula en función de los precios $\hat{y}_{i,t+k}$ los k períodos anteriores y de los atributos (variables) asociadas a las mismas exportaciones para estos k períodos anteriores, que corresponde al vector de atributos x (puerto de destino, país de destino, tipo de producto, cantidad de mercancía exportada). Adicionalmente, se contempla emplear información adicional que podría influir en el precio, descrita mediante el vector s (valores de insumos, valores en dietas, precios operativos, entre otros), valores que no se encuentran asociados directamente a x .

Este trabajo propone estudiar diferentes arquitecturas de DL con el fin de entrenar modelos para la estimación de precios y comparar los resultados obtenidos con modelos predictivos clásicos en el análisis de series temporales (ARIMA), utilizados en diferentes estudios enfocados en la estimación de precios.

Finalmente, dado que se disponen de RRAA relacionados con el comercio exterior de la salmonicultura en Chile, es posible obtener un enfoque predictivo que pueda ser implementado para la estimación de precios a nivel de productos derivados del salmón y mercados de destino, logrando medir el comportamiento del mercado y productos de mayor demanda.

1.1.2. Hipótesis

Considerando la información pública disponible y las capacidades que tienen los algoritmos de Deep Learning para lograr estimaciones más precisas en función de la arquitectura utilizada, la propuesta de tesis se basa en la siguiente hipótesis:

Utilizar modelos basados en DL para realizar estimaciones de precios de la industria salmonera chilena, según productos y mercados de destino, en función de los RRAA disponibles en SNA e integrando cuatro fuentes de información pública, permitirá elaborar modelos estimativos más precisos respecto de los modelos estadísticos clásicos que se utilizan actualmente.

1.1.3. Objetivos

Objetivo General

- Evaluar modelos de *Deep Learning* (DL) sobre datos de exportaciones nacionales provenientes desde la salmonicultura en Chile, con el fin de medir mejoras en el desempeño con respecto a modelos clásicos en la estimación de precios de salmón.

Objetivos Específicos

- OE1: Seleccionar modelos predictivos propuestos en la literatura que permitan realizar estimaciones sobre precios de salmón.
- OE2: Implementar algoritmos seleccionados basados en DL y contrastar estimaciones con algoritmos tradicionales.
- OE3: Evaluar el desempeño de los distintos algoritmos implementados según métricas y pruebas estadísticas.

1.1.4. Alcance de la Investigación

Se define un marco de trabajo para evaluar la implementación de modelos de DL en función de los datos provenientes desde el Servicio Nacional de Aduanas (SNA), información pública que no requiere procesos de anonimización o indeterminación de cifras y sin problemas de confidencialidad por parte de las empresas, que aportan de forma mensual, información sobre sus exportaciones.

Respecto de los datos, estos se publican de forma mensual y se encuentran disponibles en el portal web del SNA¹⁰ para su descarga. Por otro lado, el procesamiento del conjunto de datos, como los métodos de entrenamiento y validación y uso de diferentes modelos de DL, así como su arquitectura y selección de

¹⁰Disponible (<https://www.aduana.cl/aduana/site/edic/base/port/estadisticas.html>)

hiperparámetros, son aspectos relevantes considerados y que contribuyen en la obtención de una mejor estimación.

Finalmente, de los modelos implementados, se analiza su impacto y enfoque utilizado, a fin de lograr un modelo predictivo que pueda estimar con mayor precisión precios de salmones para distintos mercados de destino y por productos seleccionados.

1.1.5. Metodología de Trabajo

El desarrollo de la investigación contempla distintas actividades de acuerdo a la siguiente estructura de trabajo:

OE1

- Revisión bibliográfica: Revisión bibliográfica respecto de la utilización de Deep Learning y modelos tradicionales, en la estimación de precios y funcionalidades compatibles al sector en estudio.
- Estudio y selección: Estudio de enfoques de Deep Learning, identificando distintas implementaciones y soluciones en la estimación de precios.

OE2

- Implementación: Implementación de algoritmos de Deep Learning en función del problema establecido.
- Comparación: Contraste entre modelos de DL y modelos tradicionales en el uso de series temporales.
- Experimentos y ajustes preliminares: Experimentación preliminar, con el fin de ajustar parámetros y establecer adaptaciones a las implementaciones realizadas.

OE3

- Metodología de evaluación: Diseño de experimentos, estableciendo métricas de evaluación para modelos predictivos basados en datos sobre exportaciones nacionales del sector salmonicultor.
- Evaluación de resultados: Ejecución de experimentos e interpretando resultados.

Para la implementación de los modelos evaluados en esta tesis se utiliza el lenguaje de programación Python¹¹. Las bibliotecas de código abierto para la manipulación, transformación y visualización de datos, son Pandas¹², Numpy¹³ y Matplotlib¹⁴. Asimismo, en la etapa de procesado y modelamientos de datos, se emplea ScikitLearn¹⁵ y Pycaret¹⁶. Finalmente, en la implementación de algoritmos de DL se aborda con TensorFlow y Keras¹⁷.

1.2. Organización de informe de Tesis

El resto de esta tesis se encuentra estructurada por el capítulo 2 de conceptos preliminares, donde se abordan conceptos relacionados a la estimación de series temporales para el mercado de interés, detalles de métodos estadísticos en la estimación de precios, se expone una introducción sobre *Machine Learning* y conceptos relevantes en el área, se continua con una explicación sobre los modelos de *Deep Learning* que serán evaluados en la tesis. En el capítulo 3, se presentan resultados de una revisión bibliográfica seleccionando una serie de trabajos relacionados que permiten construir el estado del arte. En el capítulo 4, se detalla información relevante del conjunto de datos utilizado, preprocesamiento realizado, se explica la metodología empleada para la estimación de precios en la industria del salmón, modelos implementados y librerías de código empleadas. Luego, en el capítulo 5, se detalla el diseño de los experimentos, validación y evaluación de las capacidades del modelo, se exponen resultados de los experimentos desarrollados y

¹¹<https://www.python.org/>

¹²<https://www.pandas.pydata.org/>

¹³<https://numpy.org/>

¹⁴<https://matplotlib.org/>

¹⁵<https://scikit-learn.org/stable/>

¹⁶<https://pycaret.org/>

¹⁷<https://www.tensorflow.org/?hl=es-419>

finalmente se presenta una discusión sobre los resultados obtenidos. Finalmente, en el capítulo 6, se presentan las conclusiones del trabajo realizado y se discuten posibles trabajos futuros.

Capítulo 2

Conceptos Preliminares

2.1. Estimación de precios del salmón

2.1.1. Proceso productivo del salmón

Etapa de Agua Dulce

El proceso de Pisciculturas en agua dulce, es un proceso que dura aproximadamente 10 meses, que comienza con la incubación de las ovas (huevos), es una de las etapas más críticas en el ciclo de vida del salmón, la cual requiere de aguas claras y bien oxigenadas en condiciones de penumbra, la ova alcanza su desarrollo y rompe el huevo (eclosión), comenzando a nadar con su saco vitelino, el cual será su primera fuente de alimentación, proceso que finaliza con la absorción del saco.

Etapa de Agua Mar

La etapa de engorda en agua de mar se inicia con la recepción de los *smolts*¹ en los centros de engorda, continuando con la alimentación sostenida, por espacio de 8 a 10 meses para las especies Salmón Coho y Trucha, y de 16 a 18 meses para el Salmón Salar².

Procesamiento y Comercialización

La planta de proceso primaria recibe la materia prima (salmón o trucha entera) y mediante la utilización de tecnología y mano de obra, se lleva a cabo el faenamiento del pez a través de un proceso indoloro para él. Luego la planta de procesos transforma la materia prima de acuerdo a los requerimientos de los clientes según mercados de destino. En la Figura 2.1 se resumen las etapas del ciclo productivo del salmón.

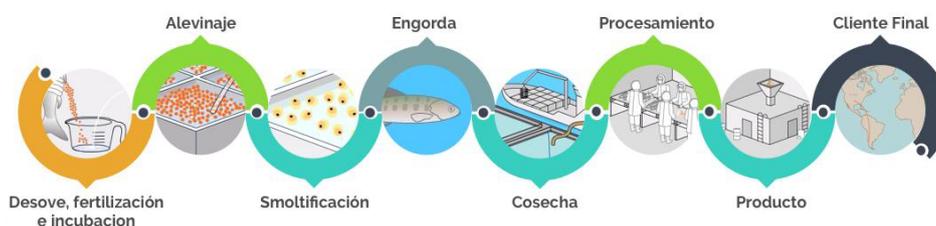


Figura 2.1: Ciclo productivo salmón (disponible en: www.salmonchile.cl).

2.1.2. Especies salmonídeas

A continuación, se detallan las principales especies de cultivo presentes en el mercado Salmonicultor de Chile³.

¹Salmón joven en la etapa de migrar desde agua dulce hacia agua mar.

²Especies contenidas en el *dataset* y definidas en Arancel Aduanero.

³<https://www.salmonchile.cl/salmon-de-chile/producto-salmonchile/>

Salmón Atlántico o Salar

Especie de mayor comercialización a nivel mundial. Es un salmón de gran tamaño y cuerpo alargado, con un exclusivo color plateado.

Salmón del Pacífico-Coho

Chile es considerado el mayor productor a nivel mundial de esta especie, que se cosecha durante los meses de septiembre y febrero de cada año. Ha sido un producto tradicional para el mercado japonés. Sus principales destinos son Japón, China, Tailandia y Corea del Sur.

Trucha Salmonídea

Al igual que el Salmón Coho, Chile es el principal productor de truchas de cultivo a nivel mundial, esta especie se caracteriza por su intenso color rojo en su carne. Adicionalmente a esto, es muy apetecida por el mercado de sushi. Los principales mercados son Japón, Tailandia, Estados Unidos (USA), Rusia y China.

En relación al mercado del salmón, la predicción de precios se basa en modelos paramétricos y en función a estos modelos, se han efectuado una serie de proyecciones del precio a nivel internacional, como las utilizada por la Bolsa de Oslo⁴ en Noruega.

2.1.3. Modelos predictivos en la estimación de precios

En ámbito de la predicción y modelización de series temporales, uno de los modelos con mejor fundamentación teórica y más utilizado es el modelo ARIMA.

Series de tiempo

Una serie temporal se define como una serie de valores o datos registrados en diferentes intervalos de tiempo. El intervalo temporal puede ser diaria, mensual o incluso anual. Por otro lado, el modelar una serie temporal para predecir sus valores futuros dependerá de una serie de factores que pueden influir su predicción utilizando técnicas paramétricas y no paramétricas al momento de generar dichas predicciones basadas en valores pasados. Uno de los modelos paramétricos más conocidos corresponde al modelo ARIMA, ampliamente utilizado para modelar series temporales y generar estimaciones.

ARIMA

Los modelos ARIMA⁵ se presentan como uno de los modelos más populares al momento de presentar estimaciones en series temporales y trabajar con variables estacionales⁶ y outliers⁷, es una versión general del modelo de media móvil autorregresiva (ARMA). En ese sentido, combina procesos autorregresivos de orden p (AR) con procesos basados en medias móviles (MA) para encontrar patrones en las series temporales y que permitan realizar estimaciones [10] y se define en la ecuación (2.1) como:

$$X_t = \alpha + \beta_1 X_{t-1} + \beta_2 X_{t-2} + \dots + \beta_p X_{t-p} + \epsilon \quad (2.1)$$

Donde:

- X_t : valor de la serie temporal en el momento t .
- α : intercepto de la ecuación.
- β_i : coeficientes de rezagos.
- p : número de rezagos (i.e. de orden AR).
- ϵ_t : error que sigue una distribución normal.

⁴<https://borsen.dk/investor/aktie/S1190/NRS/NORWAYROYALSALMON>

⁵Autoregressive Integrated Moving Average o Modelo Autorregresivo Integrado de Media Móvil.

⁶Movimientos que se producen dentro del año y que se repiten de un año a otro.

⁷Observaciones cuyos valores distan de las observaciones del mismo grupo de datos.

El modelo de media móvil con un parámetro q (es decir $MA(q)$) ilustra que el error de regresión es una combinación lineal de términos de error cuyos valores ocurrieron en varios momentos en el pasado. Significa que Y_t es una función de los errores de ϵ_t como se presenta en la ecuación 2.2.

$$X_t = \alpha + \epsilon_t + \omega_1\epsilon_{t-1} + \omega_2\epsilon_{t-2} + \dots + \omega_q\epsilon_{t-q} \quad (2.2)$$

Donde:

- X_t : el valor de la serie de tiempo en el momento t .
- α : intercepto de la ecuación.
- ω_t : coeficientes del error de rezagos.
- ϵ_t : una parte del error que sigue una distribución normal con media cero.
- q : numero de rezagos (i.e de orden MA).

Por otro lado, la función de autocorrelación parcial (PACF) y función de autocorrelación (ACF) se puede utilizar para establecer los valores p y q , respectivamente. En particular, ACF ilustra cuántos términos MA serían necesarios para eliminar la autocorrelación en una serie temporal. La parte integrada (I) ilustra que los valores de los datos de series temporales han sido reemplazados por los diferencias entre sus valores y los valores rezagados. Con el fin de generar un modelo ARIMA(p, d, q) basado en AR(p), MA(q) e I, no solo estableciendo los valores de p y q pero también estableciendo el valor de d , que es necesaria para hacer estacionarias las series temporales. El aumentado de la prueba de Dickey-Fuller (ADF) se puede utilizar para establecer el valor de d , que es el mínimo número para que los datos de la serie temporal sean estacionarios, $d = 0$ si la serie temporal ya es estacionaria.

SARIMA

La estacionalidad encontrada en una serie de tiempo pasa a ser un patrón regular de cambios que se repite durante S períodos de tiempo, donde S define el número de períodos de tiempo hasta que el patrón se repite nuevamente. El modelo ARIMA estacional (SARIMA) incorpora factores estacionales y no estacionales en un modelo multiplicativo definido como $ARIMA(p, d, q) \times (P, D, Q)_s$. Los modelos SARIMA captan el comportamiento puramente estacional de una serie, en forma similar, se realiza para la componente regular o no estacional con:

- p : orden AR no estacional.
- d : orden de diferencias no estacional.
- q : orden MA no estacional.
- P : orden AR estacional
- D : orden de diferencial estacional.
- Q : orden MA estacional.
- S : ventana de tiempo del patrón estacional.

Sin realizar operaciones, el modelo podría escribirse más formalmente de acuerdo a lo presentando en la ecuación 2.3:

$$\Phi(B^s)\phi(B)(x_t - \mu) = \Theta(B^s)\theta(b)w_t \quad (2.3)$$

Los componentes no estacionales son:

$$AR : \phi(B) = 1 - \phi_1B - \dots - \phi_p(B^p) \quad (2.4)$$

$$MA : \theta(B) = 1 - \theta_1B + \dots + \theta_q(B^q) \quad (2.5)$$

Los componentes estacionales son:

$$AR : \Phi(B^S) = 1 - \Phi_1B^S - \dots - \Phi_p(B^{PS}) \quad (2.6)$$

$$MA : \Theta(B^S) = 1 - \Theta_1B^S + \dots + \Theta_Q(B^{QS}) \quad (2.7)$$

2.2. *Machine Learning*

Machine Learning es un sub-campo de la Inteligencia Artificial como se presenta en la Figura 2.2. Ampliamente utilizado en la construcción de algoritmos que, para ser útiles, se basan en la colección de ejemplos asociados a algún fenómeno en particular, con el fin de detectar patrones significativos en un conjunto de datos, utilizando tales datos como experiencia para mejorar el desempeño de estimaciones de alguna variable de interés en nuevos casos. Es posible encontrar distintos tipos de aprendizaje basados en ML, teniendo, entre otros, el aprendizaje supervisado, semi-supervisado y de refuerzo [11][12]. El principal objetivo es desarrollar la capacidad de generalizar y asociar. Cuando esta capacidad se traduce a una computadora, significa que debería poder desempeñarse con precisión y exactitud, en tareas similares [13].

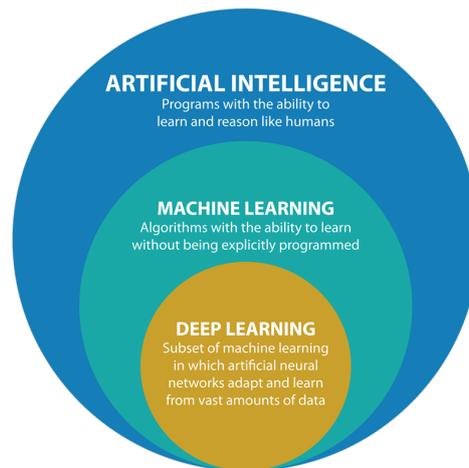


Figura 2.2: Esquema de Inteligencia Artificial y Sub-Campos (www.colab.research.google.com).

De acuerdo a lo mencionado por [14], se tienen dos criterios que ayudan a determinar si un algoritmo de ML es pertinente:

Complejidad: cuando se busca reproducir rutinas llevadas a cabo por personas como manejar, caminar, reconocimiento de voz y procesamiento de imágenes y texto, y, por otro lado, cuando se requieren encontrar patrones en grandes cantidades de datos.

Adaptividad: en los casos que se necesitan programas flexibles que se adapten a nuevos datos.

De acuerdo a los criterios antes mencionados, los algoritmos de ML han sido utilizados de forma exitosa en diversos problemas, como por ejemplo [15]:

- Clasificación de texto.
- Procesamiento de lenguaje natural.
- Reconocimiento de imágenes y de voz.
- Aplicaciones en biología computacional.
- Visión por computador.
- Detección de fraudes.
- Diagnósticos médicos.
- Sistemas recomendadores.

En función al tipo de estimaciones a realizar, es posible distinguir modelos de regresión, que busquen estimar variables continuas, y modelos de clasificación, cuyo propósito es estimar variables discretas. Entre los principales modelos de ML se pueden mencionar *Neural Networks* (NN), *Support Vector Machines* (SVM) y *Decision Trees* (DT).

Los modelos de SVM están ampliamente fundamentados en el marco de la teoría del aprendizaje estadístico, desarrollado por más de tres décadas [16], teniendo como idea base, utilizar una función denominada kernel para transformar los datos de entrada a una dimensionalidad mayor, esta dimensionalidad mayor (espacio n-dimensional), busca un hiperplano separador con el máximo de margen entre subconjuntos de instancias de distinto tipo o clase (etiqueta) [17]. Por otro lado, los modelos basados en DT son modelos de aprendizaje que generalmente son utilizados en problemas de clasificación. En esta estrategia, se divide el conjunto de datos en al menos dos conjuntos mediante una decisión respecto de los valores de algún atributo. Repitiendo esta operación se obtiene un conjunto de nodos que subdividen los datos en forma de árbol, desde su raíz (conjunto de datos completo). Los nodos internos del árbol de decisión indican una prueba de características, cada rama representa el resultado y las hojas son las decisiones (clasificaciones) tomadas tras el procesamiento [18]. Las NN son descritas en la sección 2.3, pues corresponden a uno de los métodos más empleados en la literatura, para la estimación de precios en diferentes commodities⁸, adicionalmente a esto, son la ante sala para analizar el enfoque *Deep Learning*.

El desempeño de los diferentes modelos se mide de acuerdo distintas métricas. En el caso de aprendizaje supervisado, específicamente para evaluar modelos de regresión, e igualmente utilizadas en modelos de series temporales, se comparan las estimaciones del modelo con valores conocidos del conjunto de datos, típicamente en un subconjunto separado denominado conjunto de prueba. Las métricas más empleadas para realizar las estimaciones son *Mean Absolute Error (MAE)*, *Mean Squared Error (MSE)* y *Root Mean Square Error (RMSE)*, que se describen en 2.8, 2.9 y 2.10, respectivamente, donde N representa la cantidad de instancias usadas en la evaluación.

$$MAE = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (2.8)$$

$$MSE = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (2.9)$$

$$RMSE = \sqrt{\sum_{i=1}^N \frac{(\hat{y}_i - y_i)^2}{N}} \quad (2.10)$$

Por otro lado, para evaluar modelos basados en clasificación, las métricas más empleadas son *Accuracy*, *Precision* y *Recall*, que son descritas en las ecuaciones 2.11, 2.12 y 2.13. En el escenario más común, para el cálculo de estas métricas se considera la existencia de dos clases, positiva y negativa, con el objetivo determinar los casos en que la estimación (clasificación) es acertada, si el modelo estima una clase positiva y acierta, se denomina verdadero positivo (VP); por otro lado, si estima una clase positiva y falla, se denomina falso positivo (FP); en caso de estimar una clase negativa y acierta, se denomina verdadero negativo (VN); finalmente, si la estimación pertenece a una clase negativa y falla, se denomina falso negativo (FN).

$$Accuracy = \frac{VP + VN}{VP + VN + FP + FN} \quad (2.11)$$

$$Precision = \frac{VP}{VP + FP} \quad (2.12)$$

$$Recall = \frac{VP}{VP + FN} \quad (2.13)$$

Desde el punto de vista de la evaluación del modelo, cada métrica aporta información relevante. El *Accuracy* es el número de predicciones correctas realizadas por el modelo sobre las estimaciones realizadas, *Precision* es la capacidad del clasificador de no etiquetar como positiva una instancia negativa y *Recall*, es la capacidad que tiene un clasificador de encontrar todas las instancias positiva. Finalmente, la comparación en el desempeño de diferentes modelos, se realiza en función de los puntajes obtenidos según las métricas de desempeño. Es posible complementar esta comparación con la aplicación de test estadísticos como Wilcoxon (García-Floriano et al., 2018).

⁸Materia prima. Término anglosajón que se aplica a todo producto homogéneo vendido a granel, a menudo vendidos en los mercados financieros. Los más comunes son el oro, el café, el petróleo, cobre, celulosa, etc., disponible en www.hacienda.cl

2.3. Neural Networks

El componente básico de una NN es la neurona artificial, en la que un conjunto de entradas se asigna directamente a una salida, utilizando una variación generalizada de una función lineal, esta simple instancia se denomina perceptrón, en la Figura 2.3 se presenta su arquitectura básica. En las NN multicapas, los perceptrones (también llamados neuronas) se encuentran dispuestas en capas, en donde la capa de entrada (*input layer*) y de salida (*output layer*) se encuentran separadas por un grupo de capas ocultas (*hidden layers*). Esta arquitectura por capas en la NN también se denominan *Feed Forward Networks* [19].

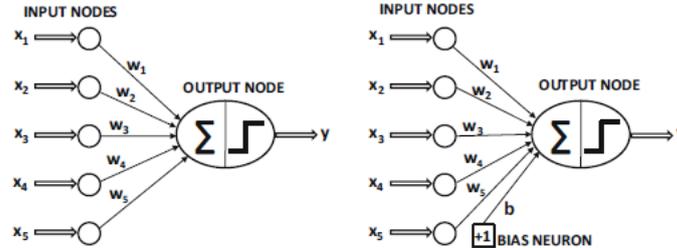


Figura 2.3: Arquitectura básica de un perceptrón (Aggarwal 2018).

Las salidas de cada neurona, resultan de la aplicación de una función de activación, esto sobre la suma de cada entrada ponderada por su respectivo peso, el concepto de estas ANNs por la interconexión inspirada en las redes neuronales biológicas [20].

Estas redes están compuestas por unidades neuronales, análogas a las neuronas de un cerebro biológico, actuando como un procesador computacional [21]. Se detalla a continuación sus componentes:

- x_1, \dots, x_n : representan las entradas de la neurona, pudiendo ser los datos de entrada, como también la conexión desde otras neuronas.
- w_0, \dots, w_n : representan el peso asociado para cada entrada (x_1 tendrá asociada un peso w_1). Esquematan la transformación de los datos de entrada. Por otro lado, w_0 representa la unidad de sesgo, valor constante que se suma a la función de activación, la cual calcula la salida de la neurona, siendo w_0 equivalente a b .
- y : salida de la neurona calculada mediante una función de activación f , función encargada de obtener dicho valor de salida, puede ser de tipo lineal, hiperbólica, etc. En 2.14, se representa la salida de la neurona obtenida desde los valores de entrada x_i :

$$y = f\left(\sum_{i=0}^n w_i x_i\right) \quad (2.14)$$

Es habitual describir la estructura de una red neuronal como un grafo, representado en la Figura 2.4, cuyos nodos son las neuronas y cada arista (dirigida) del grafo enlazado a la salida de una neurona con la entrada de otra neurona [14]. La arquitectura básica de las redes neuronales, consta de tres tipos de capas y la profundidad del modelo dependerá del número de ellas[21][22]:

- *Input layer*: consiste en un vector de atributos o *features*, que corresponde a los datos de entrada.
- *Hidden layers*: se pueden tener una o varias capas ocultas, dependerá de la complejidad que tenga el modelo.
- *Output layer*: capa final, encargada de obtener los resultados finales del modelo, en el caso de modelos de clasificación puede tener tantas neuronas como clases que puedan ser identificadas, mientras que en el caso de modelos de regresión suele ser una única neurona que genera el valor de resultado.

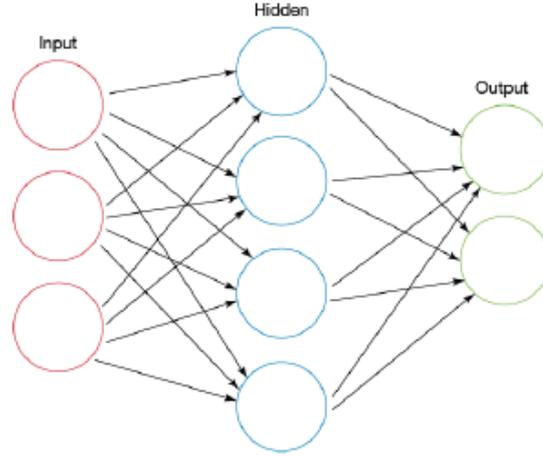


Figura 2.4: Red Neuronal Artificial (Perry 2018).

Los pesos de todas las neuronas de una capa j se pueden representar mediante una matriz W_j y los sesgos mediante un vector B_j , para un vector de entrada X_i , y empleando los pesos W_j y sesgos B_j , cada capa realiza transformaciones en los valores recibidos en cada entrada, con el fin de generar valores de salidas mediante la función de activación, lo que ocurre en cada capa hasta la capa de salida donde finalmente se genera una estimación (\hat{y}_i). Lo mencionado anteriormente puede ser representado por una función de la forma $f(X_i; W_j, B_j) = (\hat{y}_i)$. Por otro lado, al combinar varias capas y utilizar funciones de activación no lineales, es posible obtener un discriminador no lineal.

Con el objetivo de obtener estimaciones precisas, es necesario ajustar los pesos. Para lo anterior, lo primero que se requiere es comparar las estimaciones \hat{y}_i con el valor observado y_i , para cada vector de entrada X_i , lo que se calcula mediante una función de pérdida L .

La función de pérdida antes mencionada, permite determinar el costo J equivalente al error en todo el conjunto de datos para las estimaciones realizadas [23]. Esta función de costo permitirá cuantificar el error obtenido del modelo y por medio de un algoritmo como descenso por gradiente (*Gradient Descent*)⁹ es posible modificar los pesos W utilizados en las NN con el fin de disminuir la pérdida, esto, mediante el algoritmo de *Backpropagation*¹⁰ [24]. La complejidad computacional de los modelos de aprendizaje se convierte en un factor limitante en conjuntos de datos muy grandes, por tanto, con el fin de reducir este costo en el proceso de actualización, se han propuesto diversas variantes de descenso por gradiente, como por ejemplo Stochastic Gradient Descent (SGD), utilizando conjuntos reducidos de instancias $X_1, X_2, X_3, \dots, X_n$ para cada iteración [25] representada en 2.15 y 2.16 respectivamente.

$$\text{Función de Costo} = J(W) = \frac{1}{n} \sum_{i=1}^n L(\hat{y}_i, y_i) \quad (2.15)$$

$$g = \frac{1}{n'} \nabla \frac{1}{n} \sum_{i=1}^{n'} L(\hat{y}_i, y_i) \quad (2.16)$$

En la práctica, muchos problemas de baja complejidad pueden ser resueltos con NN que pueden tener dos a tres capas ocultas. A medida que son incorporadas una mayor cantidad de capas, la capacidad de capturar relaciones de mayor complejidad aumenta, dando origen a NN profundas, es decir, el enfoque de DL. Por lo tanto, los modelos de DL ofrecen soluciones a problemáticas más complejas, como por ejemplo visión por computador y procesamiento de lenguaje natural [26] y en la estimación de valores en series temporales, como la estimación de precios en el mercado de la electricidad mediante el uso de *Recurrent Neural Networks* (RNN) [27]

⁹El descenso del gradiente (Gradient Descent), es un algoritmo que estima numéricamente mediante una función, la generación de valores bajos, específicamente, encuentra valores mínimos locales. El GD, puede darnos estimaciones sin importar cuan elaborada sea nuestra función.

¹⁰El algoritmo Backpropagation determina el error generado en cada neurona, determinado por el cálculo de derivadas parciales de la función de coste, respecto a cada una de las variables.

2.3.1. Función de activación

Respecto de los experimentos y sobre las funciones de activación empleadas son linear, tanh, softmax, sigmoid y ReLU. La biblioteca *Keras* tiene disponible en un API¹¹ de funciones de activación que ayudan en su implementación. La función de activación inicial corresponde a la función lineal (Ecuación 2.13), cada experimento se inicia con esta función y por sus características, este tipo de función no efectúa modificaciones para un vector z .

$$\text{linear}(z) = z \quad (2.17)$$

Las funciones de activación *sigmoid* y *tanh* son mencionadas en la Sección 2.4.2, siendo funciones ampliamente utilizadas en RNN. *Softmax*¹² (Ecuación 2.14) convierte un vector z de n valores, en una distribución de probabilidad, tomado valores entre 0 y 1 y la suma debe dar como resultado 1. Cada vector se maneja de forma independiente. *Softmax* se utiliza generalmente como función de activación de la última capa de una red de clasificación, por que su resultado podría ser interpretado como una distribución de probabilidad.

$$\text{softmax}(z) = \frac{e^z}{\sum_j^n e^z} \quad (2.18)$$

Por otro lado, *Rectified Linear Unit* (*ReLU* es una función de activación con valores predeterminados, esto devuelve una función estándar (Ecuación 2.15), el elemento máximo de 0. La modificación de parámetros predeterminados permite utilizar umbrales distintos de cero.

$$\text{ReLU}(z) = \max(0, z) \quad (2.19)$$

2.3.2. Optimizadores

En cada modelo se consideran los optimizadores *Adam*, *SGD*, *RMSprop*, *Adadelta* y *Adamax*, optimizadores que se encuentra disponibles en la biblioteca *Keras* mediante su API¹³. Cada optimizador presenta un impacto sobre la serie de tiempo durante el proceso de entrenamiento y posteriormente, en el desempeño del modelo. Entre todos los optimizadores, *SGD*, es el más popular al momento de optimizar NN y es la base para el desarrollo de cada optimizador antes mencionado.

La figura 2.5 muestra una función de costo que puede tener varios valores mínimos locales, pero solo un mínimo global. Esto, se debe a que el descenso del gradiente avanza desde un punto en la curva y en dirección opuesta al gradiente tendera a un mínimo local. Dependiendo del valor inicial de los parámetro el descenso se puede detener en un mínimo local. Esto se puede evitar en parte con el ratio de aprendizaje y por lo mismo es un aspecto importante en los optimizadores [43].

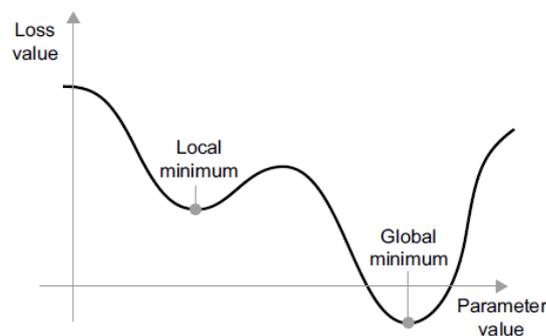


Figura 2.5: Mínimo local y mínimo global. Basado en (Chollet, 2021)

Adam (*Adaptive Moment Estimation*), es un método de descenso por gradiente estocástico, que se basa en la estimación adaptativa de momentos de primer y segundo orden. De acuerdo a (Kingma et

¹¹<https://keras.io/api/layers/activations/layer-activation-functions>

¹²<https://keras.io/api/layers/activations/softmax-function>

¹³<https://keras.io/api/optimizers/>

al., 2014) el método es computacionalmente eficiente, tiene pocos requisitos de memoria, es invariable al cambio de escala diagonal de gradientes y es muy adecuado para problemas que son en grandes términos de datos y parámetros. Por otro lado, *Adamax* corresponde a una variación de *Adam* propuestos en [44].

SGD (Stochastic Gradient Descent), dada una función diferenciable, es teóricamente posible encontrar su mínimo de forma analítica, se sabe que el mínimo de una función es un punto en que la derivada es igual a cero, pero tiene problemas cuando la superficie del espacio tiene valles, moviéndose en mínimos locales en busca del mínimo global, retrasando el proceso de entrenamiento. Adicionalmente, existen múltiples variantes de *SGD* que se diferencian por tener en cuenta actualizaciones de pesos anteriores al calcular la siguiente actualización de los pesos, en lugar de mirando el actual valor de los gradientes. Existe por ejemplo *RMSprop* y *Adadelta*, variante que se conoce como método de optimización. En particular, el concepto de *momentum*, utilizado en muchas de estas variantes, merece atención. *Momentum* aborda dos problemas del *SGD*, la velocidad de convergencia y los mínimos locales [23]. Por otro lado, en la figura 2.6, se puede visualizar el SGD a lo largo de una superficie de pérdidas. Pero no es posible visualizar el proceso real de entrenamiento de una red neuronal, debido a la dificultad de representar un espacio de 1.000.000 de dimensiones de una manera que tenga sentido para los humanos. Estas representaciones de baja dimensión pueden no ser siempre precisas en la práctica.

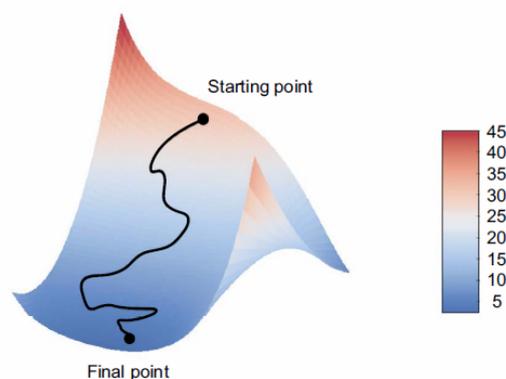


Figura 2.6: SGD pérdida de superficie. Basado en (Chollet, 2021)

2.4. Deep Learning

Deep Learning es un subcampo del *Machine Learning*. En el pasado, resultaba difícil entrenar este tipo de redes a medida que aumentaba el número de capas. Sin embargo, las implementaciones modernas de los algoritmos de aprendizaje de redes neuronales, así como el aumento en la capacidad de cómputo disponible, permiten entrenar eficazmente redes neuronales profundas [11]. A diferencia de modelos clásicos de ML, las primeras capas sirven para generar representaciones de atributos útiles para capas posteriores de forma automática.

2.4.1. Feedforward Neural Networks (FNN)

Representada en la Figura 2.7, es uno de los principales modelos de DL, utilizadas principalmente en problemas de regresión y clasificación simple. Es un tipo de NN con número variable de capas ocultas (*hidden layers*). Un caso especialmente importante de FNN es el perceptrón multicapa (MLP), en que los enlaces de la capa L_i y las salidas de cada perceptron de capa L_{i-1} son entrada para cada perceptron de capa L_i . Esta es la arquitectura FFN más utilizada. Formalmente, mediante la introducción de nodos que solo actúan para transmitir sus variables de entrada, podemos incluir la FFN más general en la clase de MLP [28].

Por otro lado, en una FNN no se presentan conexiones de retroalimentación, donde la información que ha sido procesada en forma previa se utiliza junto a nuevas entradas, lo que facilita por ejemplo modelar secuencias. Es posible mantener un estado que contenga información sobre lo ya procesado, por lo tanto, cuando una NN se le incluyen conexiones de este tipo, son llamadas *Recurrent Neural Networks (RNN)* [29].

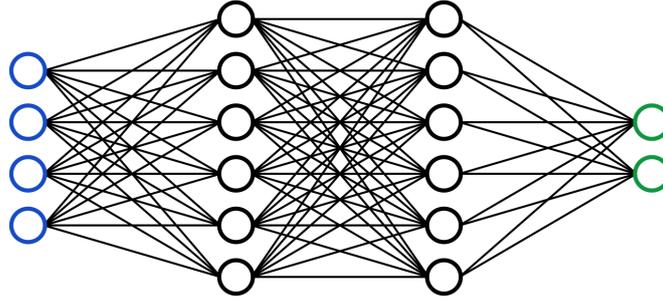


Figura 2.7: Esquema de *Feedforward Neural Networks*.

2.4.2. *Recurrent Neural Networks (RNN)*

Las RNN se utilizan para etiquetar, clasificar o generar secuencias. En general, una secuencia se define como una matriz, y las columnas que representan. Adicionalmente, este tipo de redes añade bucles de retroalimentación, consiguiendo información persistente. Se utilizan a menudo en el procesamiento de textos, debido a la estructura en formato de secuencia que estos tienen, ampliamente utilizadas en el procesamiento de voz [11][21].

Este tipo de redes procesa los datos de la secuencia por elementos y conserva un estado para representar la información que ha visto a través del paso del tiempo. La Figura 6 presenta un ejemplo de RNN utilizando la unidad recurrente básica. Este tipo de redes utiliza un vector de estado oculto H_t , que tiene la misma longitud que el número de variables de entrada, para representar la información que ha sido observada. Sus valores se establecen como ceros en el paso de tiempo inicial. El vector de estado oculto en el paso del tiempo $T - 1$ se utilizará junto con los datos de entrada en T (i.e., X_t) para calcular el estado oculto en el tiempo T . Los parámetros W_1 , W_2 y B son matrices que contienen los pesos de entrada. La función de activación utilizada se denota como $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$ o e.g., $\text{sigmoid}(z) = \frac{1}{(1 + e^{-z})}$ [23][30].

Teóricamente, cada modelo RNN (Figura 2.8), debe ser capaz de preservar la información a través del paso del tiempo. No obstante, su valor práctico para captar la dependencia a través del tiempo, suele ser limitado, esto, debido al problema de gradientes que desaparecen (Vanishing Gradients), ya que los gradientes se hacen 0 y el ajuste de pesos se vuelve muy lento. Como posible solución, se ha propuesto el modelo *Long Short-Term Memory (LSTM)* [31].

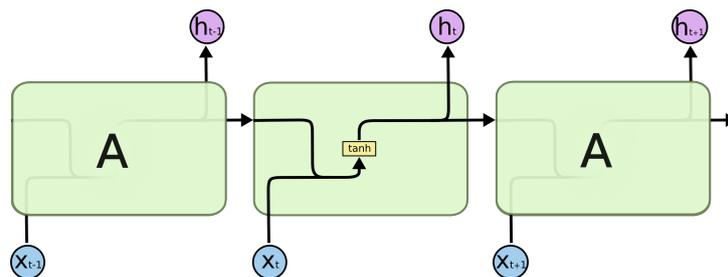


Figura 2.8: Módulo de repetición RNN de una sola capa (<https://colah.github.io>)

2.4.3. *Long Short - Term Memory (LSTM)*

Las redes LSTM son un tipo especial de RNN, capaces de aprender dependencias de largo plazo. Fueron introducidas por Hochreiter y Schmidhuber en 1997, ganando popularidad rápidamente. Una de sus principales características es que funcionan en una gran variedad de problemas y ahora son usadas ampliamente. Este tipo de redes están diseñadas para recordar información durante largos períodos de tiempo es prácticamente su comportamiento predeterminado, no es algo que les cueste aprender. Otras redes neuronales recurrentes tienen la forma de una cadena de módulos repetidos de red neuronal. En RNN estándar, este módulo repetitivo tendrá una estructura muy simple, como una sola capa de *tanh*.

Por otro lado, las LSTM también tienen una estructura similar a una cadena, pero el módulo repetitivo tiene una estructura diferente. En lugar de tener una sola compuerta (*gate*) de red neuronal, hay cuatro que interactúan de una manera muy especial como se presenta en la figura 2.9.

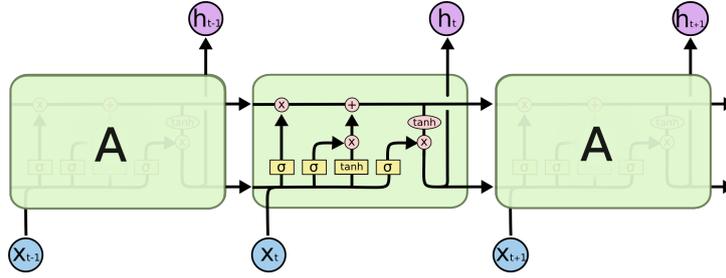


Figura 2.9: Módulo repetitivo en una red LSTM (<https://colah.github.io>)

La red LSTM es diferente a un MLP clásico. Al igual que un MLP, la red se compone de capas de neuronas y los datos de entrada se propagan a través de la red para hacer una predicción. Al igual que las redes recurrentes, las LSTM tienen conexiones recurrentes para que el estado de activaciones anteriores de la neurona al paso del tiempo anterior se utiliza como contexto para formular una salida. Pero a diferencia de otras RNN's, la LSTM tiene una formulación única que le permite evitar los problemas que impiden el entrenamiento y escalado como las RNN's [42].

La clave de las redes LSTM es el estado de la celda, su ejecución se realiza de forma directa a lo largo de la cadena, con algunas interacciones lineales, siendo fácil que la información fluya sin cambios. La red tiene la capacidad de eliminar o agregar información al estado de la celda, esto, cuidadosamente regulado por estructuras llamadas puertas. Estas puertas son una forma de dejar pasar información opcionalmente, están compuestas por una capa de red neuronal sigmoidea y una operación de multiplicación puntual.

El primer paso de cálculo en una red LSTM que se muestra en la figura 2.10, es decidir que tipo de información se va a desechar del estado de la celda. Esta decisión la toma una compuerta sigmoidea llamada "*forget gate layer*" o compuerta del olvido. Se observa h_{t-1} y x_t genera un número entre 0 y 1 para cada número en el estado de celda C_{t-1} . El valor 1 representa mantener el estado por completo, mientras que presentar un 0 es deshacerse de este estado por completo, la ecuación 2.20 representa el proceso antes mencionado

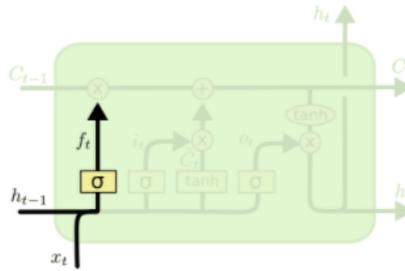


Figura 2.10: Etapa inicial de cálculo red LSTM

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \quad (2.20)$$

La siguiente etapa (figura 2.11) es decidir que información nueva será almacenada en el estado de la celda. Esto tiene dos procesos detallados en la ecuación 2.21 y 2.22. La compuerta sigmoidea llamada "puerta de entrada", decide que valores serán actualizados, luego, la compuerta *tanh* crea un vector con nuevos valores \tilde{C}_t que podría ser agregados al estado. Por otro lado, en el siguiente paso, son combinados los dos pasos anteriores para crear un estado de actualización.

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \quad (2.21)$$

$$\tilde{C}_t = \tanh(W_C * [h_{t-1}, x_t] + b_C) \quad (2.22)$$

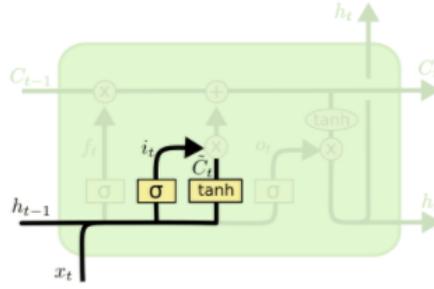


Figura 2.11: Segunda Etapa de cálculo red LSTM

Por otro lado, la siguiente etapa (figura 2.12) consiste en actualizar el estado de la celda anterior C_{t-1} al nuevo estado C_t , se multiplica el estado anterior por f_t . Luego se multiplica $i_t * \tilde{C}_t$, estos son los nuevos valores escalados, por tanto se decide actualizar cada valor de estado de acuerdo a la ecuación 2.23.

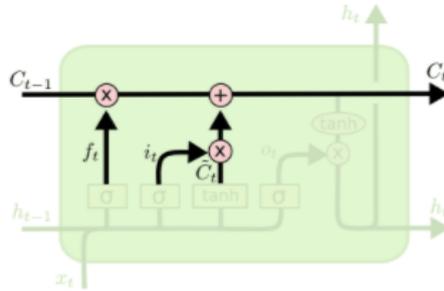


Figura 2.12: Tercera Etapa de cálculo red LSTM

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \tag{2.23}$$

Finalmente, en la última etapa (figura 2.13), la salida se basará en el estado de celda, pero se considera la versión filtrada. Primero es necesario ejecutar una capa sigmoidea que decide qué partes del estado de la celda serán generados. Luego, este estado es incorporado a la celda a través de \tanh , esto, para valores entre -1 y 1 y es multiplicado por la salida de la puerta sigmoidea, de modo que solo se emite la parte que se decide, en la ecuación 2.24 y 2.25 se presentan las etapas antes detalladas.

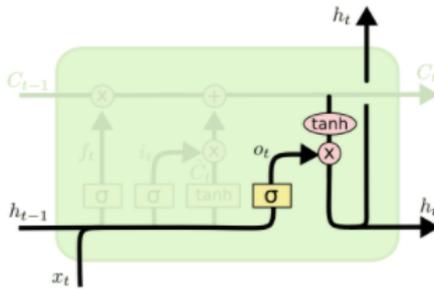


Figura 2.13: Cuarta Etapa de cálculo red LSTM

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{2.24}$$

$$h_t = o_t * \tanh(C_t) \tag{2.25}$$

2.4.4. Convolutional Neural Networks (CNN)

Otra de las principales arquitecturas en DL son las *Convolutional Neural Network* (CNN). Esto es un tipo especial de FNN, consistente en múltiples capas de conexiones neuronales con un mínimo de procesamiento sistemático [32]. Fueron utilizadas a finales de los años 90, pero durante los últimos años, el uso de éstas, se ha hecho más frecuente [13]. Las CNN han encontrado aplicaciones en el procesamiento de imágenes y procesamiento de texto, entre otras aplicaciones[11][22]. La arquitectura de las CNN's está compuesta por capas convolucionales, pooling y ReLu, capas totalmente conectadas como se muestra en la Figura 2.14. La activación ReLu no es diferente de una red neuronal tradicional, el objetivo principal de una capa convolucional es detectar bordes, líneas y otros elementos visuales como patrones locales. Esta operación matemática representa la multiplicación de vecinos locales desde un pixel específico por una determinada matriz de *kernel*. A través de los kernels, esta operación imita la extracción de características visuales como colores y bordes [32]. Los datos de entrada a la red convolucional se organizan en cuadrículas y tienen una estructura bidimensionales, los valores de los puntos de cada cuadrícula individual se denominan píxeles, y cada píxel, por lo tanto corresponde a una ubicación espacial dentro de la imagen [19].

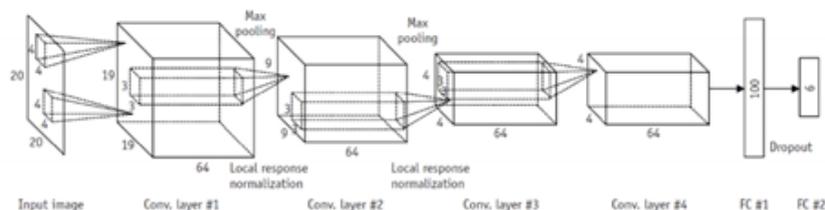


Figura 2.14: Arquitectura de CNN's (Ting 2019)

La capa convolucional utiliza una operación matemática llamada convolución, la cual desliza una función sobre otra y mide la integral de su multiplicación [46] y se define como en la ecuación 2.26:

$$s(t) = \int x(a)w(t-a) \cdot da \quad (2.26)$$

La operación denominada convolución se denota con un asterisco [47] y se detalla en la ecuación 2.27:

$$s(t) = (x * w)(t) \quad (2.27)$$

2.4.5. 1D Convolutional Neural Networks

Las CNN 1D se utilizan tanto para la extracción de características, como para la clasificación de datos, su importancia radica en las operaciones que son realizadas sobre las dimensiones de los datos, pasando de dos dimensiones (2D) siendo adaptadas para trabajar con datos 1D, teniendo especial relevancia cuando se trabaja con series temporales [43]. Por otro lado, para simplificar la analogía de la CNN, esta puede tener la libertad de cualquier dimensión de la capa de entrada independiente de los parámetros de CNN y también tener la capacidad de un solo CNN [45]. El problema presentado en esta tesis radica en un modelo de series temporales de múltiples variables que inciden en el precio final. Un modelo como CNN puede tener el potencial de generar estimaciones mediante la búsqueda de relaciones entre cada una de las instancias .

Esta arquitectura que consiste en 1D CNN aprende a extraer características mediante secuencias de observaciones y como asignar estas características internas. El beneficio de utilizar este tipo de CNN para la clasificación de secuencias es que pueden aprender directamente de los datos de series de tiempo sin procesar, y a su vez, no requieren experiencia en el dominio para diseñar manualmente las características de entrada. El modelo puede aprender una representación interna de los datos de la serie temporal e, idealmente, lograr un rendimiento comparable al de los modelos que se ajustan a una versión del conjunto del datos [42]. En el ejemplo de la figura 2.15, si tenemos una serie de tiempo de largo n y ancho k , la longitud es el número de pasos en el tiempo y el ancho es el número de variables en una serie multivariada los *kernels* de la CNN tienen el mismo ancho que la serie temporal, mientras que su longitud puede variar. De esta forma, el *kernel* se mueve en una dirección desde el comienzo de una serie de tiempo hacia su final, realizando una convolución. No se mueve a la izquierda ni a la derecha como lo hace cuando se aplica la convolución 2D habitual a las imágenes.

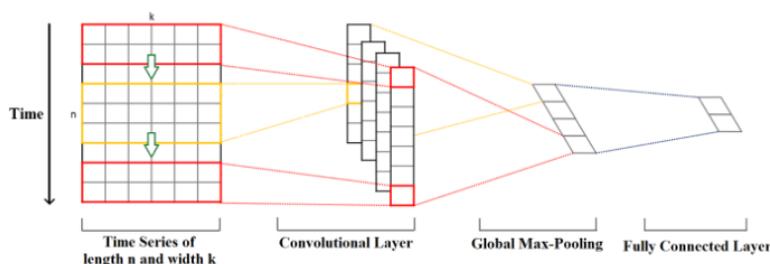


Figura 2.15: 1D CNN para series de tiempo (Kim 2014).

Los elementos *kernels* se multiplican por los elementos correspondientes de la serie temporal que cubren en un punto dado. Luego, los resultados de la multiplicación se suman y se aplica una función de activación no lineal al valor. El valor resultante se convierte en un elemento de una nueva serie de tiempo univariante filtrada y luego el kernel avanza a lo largo de la serie de tiempo para producir el siguiente valor. El número de nuevas series temporales filtradas es el mismo que el número de *kernels* de convolución. Dependiendo de la longitud del kernel, diferentes aspectos, propiedades y características de la serie temporal inicial se capturan en cada una de las nuevas series filtradas. El siguiente paso es aplicar la agrupación *max-pooling* a cada uno de los vectores de series temporales filtrados: se toma el valor más grande de cada vector, se forma un nuevo vector a partir de estos valores, y este vector de máximos es el vector de características final que se puede usar como entrada para una capa regular completamente conectada [48].

Capítulo 3

Estado del Arte

En materia de estimación de precios es importante conocer las distintas soluciones propuestas en la literatura relativas a la estimación de precios para el mercado del salmón, profundizando aspectos importantes que pueden ser útiles para llevar a cabo la tesis propuesta. Una revisión del estado del arte permite conocer los avances actuales en la materia propiciando el desarrollo de nuevas y más alternativas.

Esta tesis contempla evaluar modelos de DL teniendo por objetivo desarrollar un modelo predictivo más preciso que los modelos ARIMA en la materia de estimación de precios para el mercado del salmón. En las búsquedas iniciales no se encontraron trabajos relacionados con la estimación de precios para el salmón utilizando modelos de DL, pero sí en la utilización de modelos paramétricos como lo es ARIMA. En relación a lo expuesto, se amplía la búsqueda para trabajos que apliquen métodos de ML para la estimación de precios en el mercado del salmón.

3.1. Proceso de revisión

La búsqueda de trabajos relacionados se establece a partir de un string de búsqueda inicial, para ir mejorando en relación a términos extraídos de los diferentes documentos encontrados. El proceso iterativo se inició con bajo el siguiente string de búsqueda: "*(Salmon Market OR Salmon Price OR Salmon Cost Transfer OR Salmon Exports) AND (Estimation OR Prediction OR Forecasting) AND (Machine Learning OR Deep Learning OR Neural Networks) AND NOT (Salmon Farming OR Salmon Growings)*".

Las fuentes de búsqueda utilizadas en la revisión corresponden a las siguientes bases de datos electrónicas: IEEE¹, ACM² y Web of Science³. En base a los términos clave se diseñaron diferentes vectores de búsqueda permitiendo optimizar la calidad y cantidad de resultados.

Como criterio de inclusión fueron considerados todos los trabajos que se referían a la estimación de precios en el mercado nacional e internacional del salmón mediante la aplicación de modelos de ML y DL. Por otro lado, como criterio de exclusión, se descartaron aquellos trabajos que (1) no aplican modelos de ML y DL en la estimación de precios de salmón, (2) consideraban enfoques paramétricos en procesos estimativos (3) presentaban estimaciones basados en volúmenes de cosecha (4) consideraban análisis farmacológicos, (5) proyecciones sobre consumo de dietas. Aplicados los criterios descritos anteriormente, fueron seleccionados 2 artículos resumidos en el Cuadro 3.1.

Una gran cantidad de trabajos fueron descartados por abordar estudios o procesos que no se ajustaban a la estimación de precios o no incorporaban modelos de ML y DL en sus análisis. Trabajos que buscaban determinar el comportamiento de dietas, resistencia hacia algún virus en particular que pueda afectar la biomecánica de los salmones.

¹<https://ieeexplore.ieee.org/>

²<https://dl.acm.org/>

³<https://www.webofknowledge.com>

3.2. Resultados

3.2.1. Estimación de precios utilizando modelos de ML y DL en el mercado del salmón

De los trabajos seleccionados, en [3] se realiza un estudio comparativo de diferentes modelos econométricos y de ajuste estacional, para la estimación de precios en el mercado de salmón noruego y al mismo tiempo, la utilización de NN y k-nearest neighbors method (kNN) como técnica de clasificación y regresión. Como métricas de evaluación se utilizó RMSE, MAE, MAPE⁴ y MASE⁵. Los datos utilizados en este análisis corresponden a compañías productoras y exportadoras de salmón noruego con participación en la Bolsa de Oslo.

Por otro lado, en [33] se analiza la eficiencia y aplicación de métodos para la estimación de precios de salmón atlántico, utilizando como datos de entrada la información proporcionada por Fish Pool Price⁶, examinado el comportamiento y precisión de las predicciones, utilizando modelos sobre ajuste estacional, Unit Root Testing⁷, test ADF⁸, criterios AIC⁹ (Akaike Information Criterion) y como métrica de evaluación se utilizó EMC.

Cuadro 3.1: Resumen de trabajos relacionados en la estimación de precios en el mercado del salmón

Ref.	Año	Problema	Algoritmo/Enfoque	Métrica (s)	Datos
[3]	2017	Estimación de precios de salmón	Seasonal adjustment, Univariate ARIMA, VAR, Artificial neural network & k-nearest neighbors method	RMSE, MAE, MAPE, MASE	Corporaciones productoras y exportadoras pertenecientes a la Bolsa de Oslo,
[33]	2019	Eficiencia en la estimación sobre el mercado del salmón atlántico	AIC, ADF, Unit Root Testing, NN	EMC	Fish Pool Price

3.2.2. Estimación de precios utilizando modelos de ML y DL

Por otro lado, se estableció un segundo criterio de búsqueda con enfoque en ML y DL para la estimación de precios en distintos sectores económicos, la cadena de búsqueda, tiene como enfoque, la estimación de precios de diferentes mercados y/o productos: "(Price Market OR Time Series) AND (Forecasting OR Estimation OR Prediction) AND (Machine Learning OR Deep Learning OR Neural Networks)". Los motores de búsquedas son aquellos mencionados con anterioridad. La necesidad de un segundo criterio de selección de artículos responde a la falta de información con enfoque en ML y DL para la estimación de precios en el mercado del salmón a nivel nacional e internacional.

Los criterios de inclusión considerados durante esta segunda búsqueda consideraban aquellos trabajos que hacían referencia a la aplicación de modelos de ML y DL para estimar precios en diferentes mercados y sectores productivos. Los criterios de exclusión empleados descartaron aquellos trabajos que (1) no aplicaban modelos de ML y DL para estimar precios, (2) Utilizan técnicas paramétricas en la estimación de precios (3) no consideran estimación de precios. Tras la aplicación de los criterios descritos con anterioridad, se seleccionaron 9 artículos resumidos en el Cuadro 3.2.

Cabe destacar que, la inclusión de una segunda revisión de artículos aportó una mayor cantidad de resultados e información sobre estimar precios utilizando modelos de ML y DL. Si bien es cierto, los

⁴Mean Absolute Percentage Error o Error Porcentual Absoluto Medio, es una medida de precisión de predicción de un método estadístico y que mide el desempeño del pronóstico.

⁵Mean Absolute Scaled Error o Error en Escala Absoluta Media corresponde al error absoluto medio de los valores pronosticados dividido por el error absoluto medio.

⁶Bolsa Europea de transacciones financieras sobre salmón. Disponible en <https://fishpool.eu/>

⁷Contraste de Dickey-Fuller que detecta estadísticamente la presencia de conducta tendencial estocástica en series temporales. La hipótesis nula de esta prueba es que existe una raíz unitaria en la serie.

⁸Prueba de Dickey-Fuller aumentada, es una prueba de raíz unitaria. Su versión aumentada es utilizada para conjuntos de datos más amplios y modelos de mayor complejidad en series de tiempo.

⁹El criterio de información de Akaike es una medida de la calidad relativa de un modelo estadístico, para un conjunto dado de datos. El AIC proporciona un medio de selección del modelo.

resultados no se enfocaron en el mercado del salmón, si los modelos implementados servían como base para generar un modelo predictivo en función a los datos que se utilizarán en la implementación.

Cuadro 3.2: Resumen de trabajos relacionados en la estimación de precios

Ref.	Año	Problema	Algoritmo/Enfoque	Métrica (s)	Datos
[6]	2019	Estudio comparativo en la predicción de precios de Bitcoin utilizando Deep Learning	CNN, LSTM, CRNN, Ensemble, ResNeT, SVM, Base, Random	MAPE, Accuracy	Bitstamp Bitcoin Price (USD)
[5]	2018	Estimación de precios en el mercado de las acciones utilizando herramientas de Machine Learning y Deep Learning	MFNN, RNN, CNN, LSTM, Linear Regression, Logistic Regression, Random Forest, SVM	R, AR, SR, V, MDD, DWR	30 datasets, sin detallar procedencia
[34]	2019	Estimación del Precio Medio utilizando Deep Learning	MLP, CNN, RNN, LSTM	RMSE	Amazon, Google, Nordic datasets
[35]	2018	Predicción de Series de Tiempo y Comparación con Modelos no tradicionales	NN, Holt Winters, AR	AUC	Precio de retail y precio de ventas de tomates en USA (Enero 1960-Diciembre 1999)
[36]	2018	Comparación de diferentes modelos de Deep Learning en la estimación de series de tiempo	Random Forest, CNN, LSTM, MLP, CNN-LSTM	RMSE	Datos meteorológicos
[37]	2021	Análisis experimental de diferentes arquitecturas de DL en la estimación de series de tiempo	RNN, LSTM, CNN, MLP, GRU, TCN, ESN	MAE, WAPE	CIF2016o12, CIF2016o6, ExchangeRate, M3, M4, NN5, SolarEnergy, Tourism, Traffic, Traffic-metr-la, Traffic-perms-bay, WikiWebTraffic
[9]	2021	Uso de arquitecturas basadas en DL para la estimación en series de tiempo	RNN, Hybrid Models, LSTM	-	-
[38]	2019	Uso de hiperparámetros en modelos de Machine Learning con enfoque en la predicción de series temporales	RNN, LSTM, MLP, CNN	MAPE	M4 Kaggle dataset
[39]	2020	Estudio comparativo sobre distintos dataset aplicando diferentes modelos en la estimación de series temporales	EWT, LSTM, ORELM	MAPE, RMSE, MAE, SDE	data of the SP 500 from 17, Dec 2010 to 17, Jan 2013 - data CMSB from 18, Dec 2013 to 18, Jan 2016

En [6] los investigadores realizan un estudio comparativo entre diferentes arquitecturas de DL, RNN, CNN y LSTM aplicado sobre el mercado de precios de Bitcoins. Los autores destacan que RNN y LSTM presentan un mejor desempeño en la estimación de precios, por sobre lo presentado por modelos ARIMA.

En [5] destacan que estimar precios de acciones de diferentes mercados presenta grandes dificultades para los investigadores y especuladores de precios. Por tanto, utilizar modelos de DL en procesos de predicción de precios de acciones presenta mejores rendimientos a nivel de estimación. En este estudio, los investigadores realizan análisis comparativos de diferentes arquitecturas basadas en DL (CNN, RNN, LSTM), como también modelos de SVM, Random Forest, modelos de regresión lineal, regresión logística KNN, NN, con el objetivo de modificar hiperparámetros en los distintos modelos, a fin de obtener estimaciones más precisas y modelos más eficientes. Se destacan a RNN con resultados más estables al momento de estimar precios en el mercado de las acciones.

En [34] los autores destacan que, el mercado financiero ha incrementado su complejidad en el proceso de automatización y análisis de información, destacando que métodos tradicionales han fallado al momento de estimar el mercado de valores. Por otro lado, los investigadores destacan que modelos de ML y DL presentan mecanismos más efectivos al momento de predecir y analizar series de tiempo comparados con modelos tradicionales como ARIMA.

En [35] los autores demuestran que utilizando métodos como NN en la predicción de precios, es posible obtener mejores estimaciones por sobre los clásicos métodos estadísticos.

En [36] los autores establecen comparaciones de diferentes modelos de DL, Random Forest, CNN, LSTM, MLP, modelos híbridos CNN-LSTM, en la estimación de series de tiempo, enfocado en datos meteorológicos, destacando el uso de FFN en la estimación de problemas específicos (precipitaciones, consumo de electricidad y quiebra de empresas). Adicionalmente a esto, destacan la integración de modelos de NN y ARIMA para series de tiempo con el fin de obtener mejores resultados. El proceso de evaluación de los distintos modelos se llevó a cabo mediante RMSE, destacando mejores resultados mediante modelos híbridos CNN-LSTM por sobre modelos individuales.

En [37] los investigadores establecen un estudio comparativo de diferentes arquitecturas de DL en la estimación de series de tiempo, haciendo énfasis en la optimización de parámetros con el objetivo de obtener modelos más precisos, destacan además que modelos basados en CNN 1D y LSTM presentan mejores alternativas a la hora de abordar problemas de series de tiempo. Se hace mención a LSTM obteniendo mejores resultados al trabajar con series de tiempo más acotadas y, por otro lado, CNN presenta un mayor rendimiento de cómputo y precisión en las estimaciones, recomendando series de tiempo más extensas.

En [9] los investigadores destacan el uso de diferentes arquitecturas de DL y su adaptabilidad a problemáticas de series de tiempo, adicionalmente a esto, señalan que gracias a la disponibilidad de datos y la creciente potencia informática desarrollada en los últimos tiempos, las de arquitecturas DL han logrado tener éxito en la predicción de series temporales.

En [38] los autores experimentan la precisión en la predicción de series temporales para distintos tipos de arquitecturas basadas en NN, RNN, LSTM y CNN 1D, modificando hiperparámetros con el fin de conseguir modelos con mejores rendimientos al momento de generar estimaciones.

En [39] los investigadores analizan el uso de DL en la estimación de precios en el mercado de las acciones, utilizando diferentes tipos de arquitecturas, evaluando cada una de estas implementaciones según MAPE, destacando a los modelos híbridos por sobre modelos individuales. Los trabajos expuestos y que han sido desarrollados en el último tiempo, muestran que los algoritmos basados en DL son una herramienta potente al momento de generar estimaciones sobre series temporales, presentando mejores rendimientos por sobre los modelos clásicos utilizado en este tipo de problemas (ARIMA, SARIMA, entre otros). Cabe señalar que, las arquitecturas de DL más utilizadas en para la estimación de series temporales, destacan RNN, CNN 1D y LSTM.

Finalmente, es importante señalar que de los artículos revisados, solo en [3] se utilizan modelos de ML para la estimación de precios en la salmonicultura, y en [33] se realizan estimaciones de precios en este sector económico utilizando métodos tradicionales.

3.3. Síntesis

La estimación de precios en el ámbito de la salmicultura es un problema que puede ser abordado con modelos ML y DL y puede tener una gran incidencia en la planificación de recursos, como también estrategias de mercado, desarrollando políticas públicas que pudieran aportar al desarrollo del sector. Destacan dos trabajos con enfoque en el mercado del salmón a nivel mundial aplicando técnicas de ML y otros modelos paramétricos (ARIMA) utilizados ampliamente para estimar precios de distintos mercados, pero no se logró identificar la implementación de modelos de DL para estimar precios del mercado nacional e internacional del salmón.

Capítulo 4

Implementación

Este capítulo detalla las etapas llevadas a cabo para realizar el entrenamiento y evaluación de modelos de DL para estimación de precios de salmón, incluyendo un análisis y preparación del conjunto de datos disponibles. Se describe el preprocesamiento realizado, los distintos conjuntos de datos utilizados y las herramientas empleadas para la implementación de los distintos modelos de *Deep Learning*.

4.1. Conjuntos de datos

El proceso se inicia con la exploración y tratamiento de un conjunto de datos que describen transacciones de exportación de 99.353 instancias, descritas por 19 variables o atributos, obtenido desde el Servicio Nacional de Aduanas (SNA¹). Durante la etapa de preprocesamiento de los datos, se diseñaron diferentes series de tiempo para implementar modelos predictivos. Adicionalmente a esto, y en función al planteamiento del problema, se añadieron atributos adicionales a las series de tiempo incluyendo información como Índice de Precios al Consumidor (IPC), tipo de cambio (como mecanismo de transmisión en los valores de insumos y costos de las empresas), valor internacional del petróleo, valor internacional de aceite de soya, IPC Estados Unidos (USA), IPC Eurozona. Es importante destacar que la incorporación de variables externas al conjunto de datos original, radica en la importancia que tienen dichos atributos en la variación del precio de comercialización a nivel internacional. Todas las variables del conjunto de datos principal se describen en el Cuadro 4.1 y que contiene información proveniente de diferentes regiones en Chile, detallando diferentes mercados de destino, y diferentes tipos de productos, descritos según códigos arancelarios.

La figura 4.1 se muestra la estructura completa del conjunto de datos original, como también aquellas variables de fuentes externas, utilizadas para robustecer cada uno de los modelos y que también explican el comportamiento económico del precio del salmón.

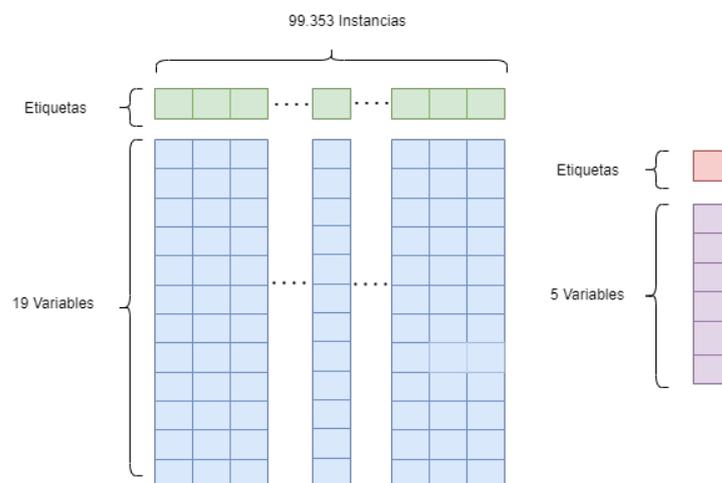


Figura 4.1: Conjuntos de datos y su composición

¹<https://www.aduana.cl/aduana/site/edic/base/port/comex.html>

Cuadro 4.1: Variables del conjunto de datos obtenidos desde el Servicio Nacional de Aduanas (SNA)

Variable	Tipo	Descripción
Período	numérica	Año de exportación
Mes	numérica	Mes de exportación
Cod aduana tramitación	numérica	Aduana de tramitación, hace referencia a frontera de salida
Cod tipo operación	numérica	Operación de salida, exportación, manifiesto de salida
Cod region origen	numérica	Región de origen carga
Cod via transporte	numérica	Tipo de transporte
Cod puerto embarque	numérica	Puerto de embarque exportaciones
Cod puerto desembarque	numérica	Puerto de desembarque exportaciones
Cod pais destino	numérica	País destino carga
Cod modalidad venta	numérica	Tipo de venta utilizada en la transacción
Item sa	numérica	Código arancelario
Fob us dusleg	numérica	Valor de la mercancía estipulada en DUS (Declaración Única de Salida)
Fob us ivv	numérica	Valor de la mercancía estipulada
Peso bruto (kg)	numérica	Peso bruto carga
Cantidad mercancía	numérica	Cantidad de mercancía exportada
Cod unidad medida	numérica	Unidad de medida utilizada en carga transportada
Moneda	numérica	Tipo de moneda utilizada en transacción
Cláusula venta	numérica	Cláusulas de compra y venta

Por otro lado, en el Cuadro 4.2 se detallan cada uno de los atributos adicionales, que serán incorporados a las series de tiempo que servirán como entrada al modelo predictivo.

Cuadro 4.2: Conjunto de datos adicionales para el modelamiento

Atributo	Descripción	Fuente
s ₁	Precios petróleo	https://www.worldbank.org/en/research/commodity-markets
s ₂	Precios aceite de soya	https://www.worldbank.org/en/research/commodity-markets
s ₃	US Índice IPC	https://www.bls.gov/gov/cpidata
s ₄	Europa IPC	https://ec.europa.eu/eurostat/data/database
s ₅	Tipo de cambio	https://www.bcentral.cl/web/banco-central/areas/estadisticas/tipos-de-cambios-y-paridades

4.2. Preprocesamiento de datos

Como primera etapa, el preprocesamiento de los datos es una tarea relevante. Es necesario seleccionar las variables de interés provenientes desde el conjunto de datos original y mantener la información que permita entrenar cada uno de los modelos a evaluar.

El análisis preliminar del conjunto de datos original permitió identificar la necesidad de construir la variable *price*, la cual no está contenida en el conjunto de datos original. Si bien es cierto, el conjunto de datos contiene las variables Fob us dusleg², esta no refleja el valor de mercado (US/Kg) que tiene el salmón para cada país de destino.

Del conjunto de datos original fueron seleccionadas las variables de interés que aportaban sentido económico y las fluctuaciones de mercado que tiene el salmón en cada destino, según tipo de producto. Por otro lado, fueron descartadas aquellas variables que no aportan a la estimación de precios y que tienen sentido económico para explicar el comportamiento del mercado como por ejemplo tipo de carga, comuna de procedencia, tipo de contrato, atributos que por sus características no influyen en el precio de venta del salmón. Las variables seleccionadas del conjunto de datos original para la construcción de las series temporales se detallan a continuación.

- Período.
- Mes.
- País.
- Item-SA.
- Fob us dusleg.
- Peso-Bruto (kg)

VARIABLES NECESARIAS PARA LA CONSTRUCCIÓN DE LAS SERIES DE TIEMPO A UTILIZAR EN LOS DIFERENTES MODELOS DE DL A IMPLEMENTAR.

4.2.1. Construcción variable *price*

Como se mencionó anteriormente, en la construcción de la variable *price* necesaria para su incorporación al modelo de DL, se utilizan las siguientes variables del conjunto de datos original:

- Fob us dusleg.
- Peso-Bruto (kg)

Para la construcción de la variable precio (*price*), que será nuestra variable a estimar se utiliza la siguiente relación detallada en la ecuación 4.1:

$$P_t = \frac{p_t}{Q_t} \quad (4.1)$$

Donde:

- P_t : corresponde al precio en un tiempo t .
- p_t : corresponde al valor de exportación en el mercado de destino en un tiempo t (Fob us dusleg).
- Q_t : corresponde a la cantidad exportada en un tiempo t (Peso-Bruto (kg)).

En la figura 4.2, se presentan los estadísticos para la variable FOBUS. Se observa un sesgo en la distribución y permite visualizar con claridad que, en general, los valores de exportaciones hacia diferentes mercados de destino desagregados por código arancelario, en el conjunto de datos utilizado, son bastante bajos.

En la figura 4.2 se puede observar en histograma de la variable Peso-Bruto(Kg) utilizada en la construcción de la variable precio (*price*) y se presentan estadísticos generales y al igual que la descripción de la variable anterior, se observa un sesgo en la distribución y se puede observar que el peso bruto medido en kilogramos (Kg) de las exportaciones tiene relación directa en el precio de exportación del producto según su mercado de destino, explicado por la relevancia que tienen ciertos productos derivados del Salmón Salar, siendo el salmón fresco y refrigerados los más importados.³

²Valor de una mercancía de exportación cuando está embarcada en la nave, antes de salir hacia su destino.

³Estudio de Mercado de Salmón Chileno en Estados Unidos 2018, disponible en: <https://acceso.prochile.cl/wp->

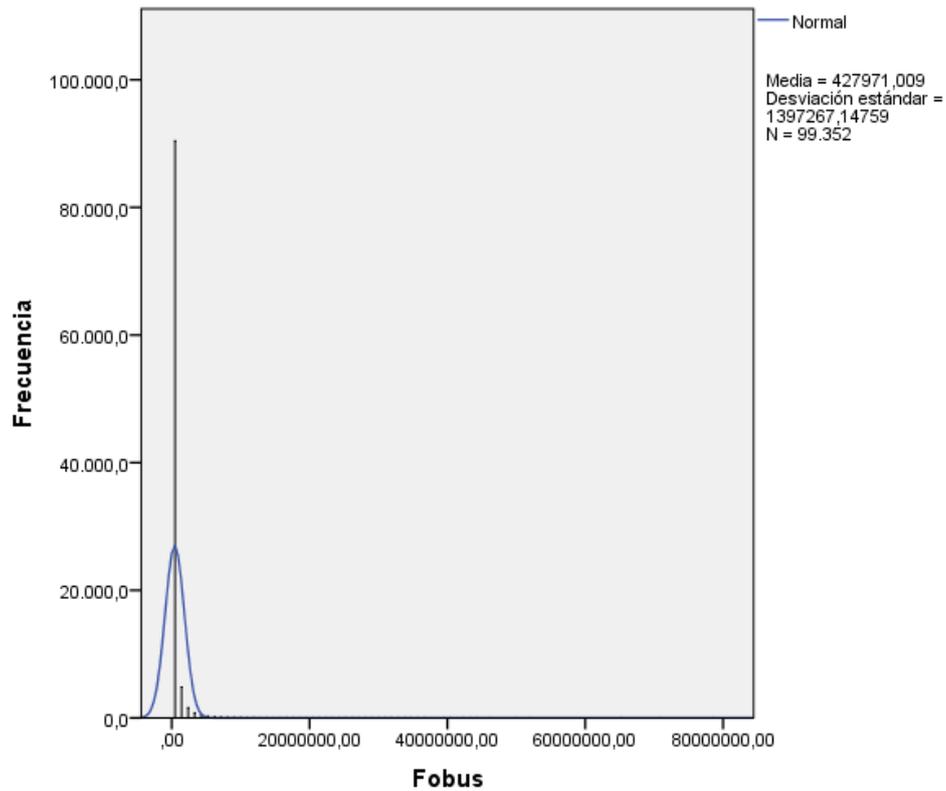


Figura 4.2: Histograma para la variable fobus

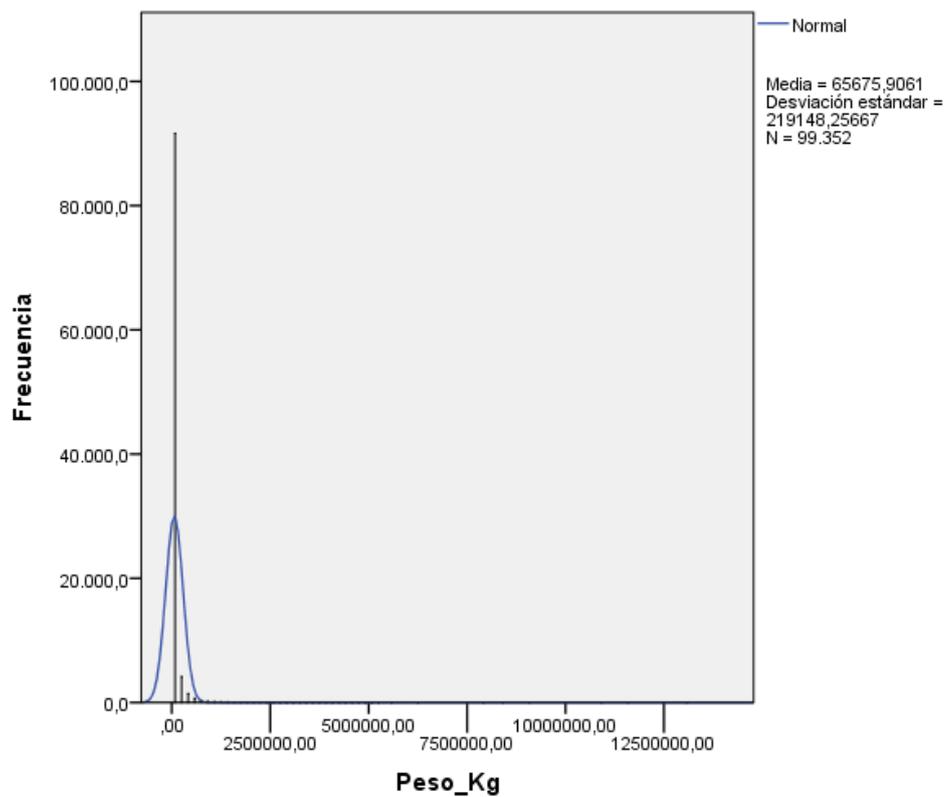


Figura 4.3: Histograma para la variables peso bruto (kg)

4.2.2. Diseño y elaboración de series temporales

Diseñada la variable a estimar (*price*) se elaboran las series temporales a utilizar. Dado que en el dataset se cuenta con información correspondiente a 128 mercados de destino, es necesario definir un número reducido de mercados relevantes para realizar la experimentación, lo que corresponde a ciertos valores en la variable País de destino, en este caso, se utilizarán como destino Estados Unidos y Europa en su conjunto debido a la relevancia que tienen en las exportaciones de salmón y como el salmón de origen chileno ha ingresado en estos mercados. Del mismo modo, es necesario definir un conjunto acotado de tipos de productos a analizar y que se obtiene de acuerdo a los códigos arancelarios según Item-SA, establecidos por el SNA. En este caso, el análisis se enfocará en salmón salar por la relevancia que tiene en la producción a nivel nacional (descrito en sección 2.1), y al mismo tiempo, considerando el tipo de exportación de este producto para los mercado de destino, tanto para Estados Unidos como para Europa, se diseñan series temporales para productos congelados y productos fresco. Lo que se busca advertir aquí es el precio de mercado de este tipo de productos y el impacto que tiene el transporte de un producto fresco hacia un destino en particular, debido al poco tiempo que este tiene para llegar a su destino, lo cual puede encarecer este tipo de producto. En este caso, es importante definir de forma correcta los códigos arancelarios con el objetivo de construir series de tiempo con cada una de las definiciones establecidas. Identificar correctamente los códigos que corresponden a los productos definidos no se presentó como una simple tarea, este proceso de revisión consto con 50 códigos⁴ para el desarrollo de cada una de las series de tiempo.

Finalmente, a partir de la selección de variables, definición de códigos arancelarios, mercados de destino e incorporación de variables contenidas en el cuadro 4.2 se elaboran las siguientes series temporales que serán empleadas para entrenar y evaluar los distintos modelos:

- Serie Salmón Salar: Serie de tiempo de precios del mercado internacional de salmón salar (incluye valores frescos y congelados), no hace distinción según mercado de destino.
- Serie Salmón Salar USA: Serie de tiempo de precios salmón, con destino al mercado de Estados Unidos (incluye valores frescos y congelados).
- Serie Salmón Fresco USA: Serie de tiempo de precios de salmón en formato fresco, con destino al mercado de Estados Unidos.
- Serie Salmón Congelado USA: Serie de tiempo de precios de salmón en formato congelado, con destino al mercado de Estados Unidos.
- Serie Salmón Fresco Europa: Serie de tiempo de precios de salmón en formato fresco, con destino a la Euro zona (no se hace distinción de un país en particular).
- Serie Salmón Congelado Europa: Serie de tiempo de precios de salmón en formato congelado, con destino a la Euro zona (no hace distinción a un país en particular).

Todas las variables contenidas en cada una de las series temporales (se incluye variable *price*)son normalizadas de forma individual, esto, empleando promedio y desviación estándar de acuerdo a lo presentado en la Ecuación 4.2, donde la variable normalizada z es el resultado de la resta del promedio μ al valor de cada variables x , dividido por su desviación estándar σ . Este proceso de normalización de los valores de cada una de las series temporales permitirá suavizar valores atípicos y que no afecte la serie al momento de generar estimaciones, manteniendo una misma escala para cada una de las instancias de forma simétrica.

$$z = \frac{x - \mu}{\sigma} \quad (4.2)$$

⁴Ver Anexo 1

En resumen, del preprocesamiento de los datos, se llevaron a cabo las siguientes actividades o tareas:

- Análisis de valores ausentes.
- Selección/exclusión de variables de interés.
- Construcción variables precio.
- Definición mercados de destino a utilizar.
- Definición código arancelarios.
- Construcción de series temporales.
- Normalización de valores de series temporales.

A partir del conjunto de datos de 99.353 instancias se diseñaron series de tiempo de diferentes tamaños y horizontes de tiempo. Por un lado, información contenida para productos frescos y congelados para mercados de Estados Unidos y Euro zona contaban con poca información en el dataset original, de acuerdo a su definición obtenida de los códigos arancelarios. Esto, se presenta como una oportunidad para comparar el rendimiento de modelos con series de tiempo de diferentes tamaños y poder establecer un modelo predictivo más robusto. Para las series de tiempo Salmón Salar y Salmón Salar USA se estableció un período de inicio desde el año 2002, con 237 instancias para ser incorporadas al modelo de DL. Por otro lado, Para las series de tiempo, Salmón Salar Congelado y Salmón Salar Fresco, el inicio se estableció a partir del año 2012, con 117 instancias para su incorporación al modelo de DL. Finalmente, las series de tiempo Salar Fresco Europa y Salar Congelado Europa su período de inicio es el año 2012 totalizando 117 instancias. El cuadro 4.3 presenta información resumen en función a lo detallado anteriormente.

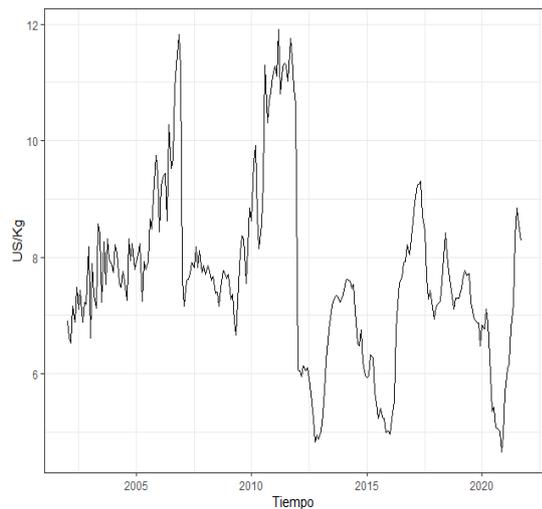
Cuadro 4.3: Series de tiempo períodos de inicio y término

Serie de Tiempo	Período inicial	Período final
Serie Salmón Salar	Enero de 2002	Septiembre de 2021
Serie Salmón Salar USA	Enero de 2002	Septiembre de 2021
Serie Salmón Salar Fresco USA	Enero de 2012	Septiembre de 2021
Serie Salmón Salar Congelado USA	Enero de 2012	Septiembre de 2021
Serie Salmón Salar Fresco Europa	Enero de 2012	Septiembre de 2021
Serie Salmón Salar Congelado Europa	Enero de 2012	Septiembre de 2021

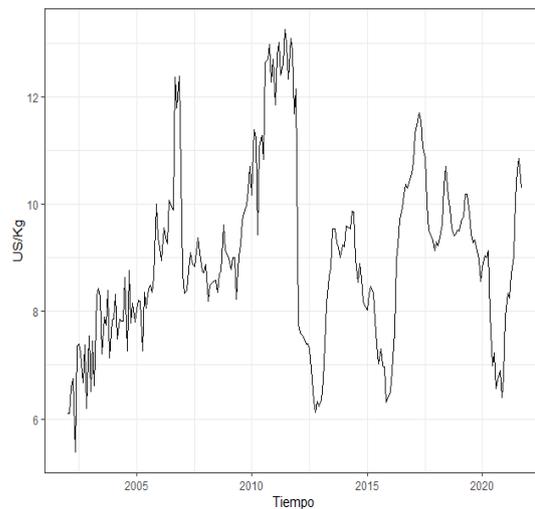
A continuación se presentan las series detalladas anteriormente y su comportamiento a través del tiempo, la figura 4.4 presenta la evolución del precio para el mercado de salmón salar a nivel mundial.

En particular, aquellas series de tiempo de menor longitud pueden presentar dificultades al momento de generar estimaciones, tanto para los modelos de ML y DL, presentando errores en sus estimaciones que puedan dificultar su interpretación.

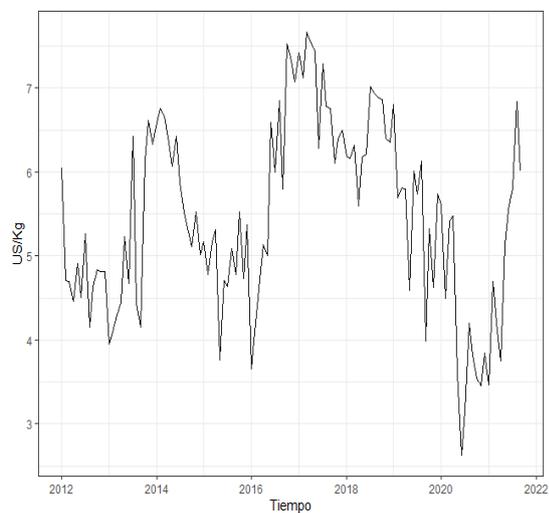
En síntesis, se cuenta con seis series temporales de diferente tamaño que describen la evolución de precios de salmón, junto a información complementaria definida en el cuadro 4.3, a fin de robustecer los modelos predictivos a evaluar. Cada serie contiene información asociada a los mercados de Estados Unidos y Europa según corresponda, como el Índice de Precios al Consumidor (IPC), y otros atributos que impactan directamente sobre el precio del salmón a nivel mundial y que son generados por cambios monetarios como es el precio del petróleo y del aceite de soya (insumo importante para la generación de las dietas de los salmones en crecimiento).



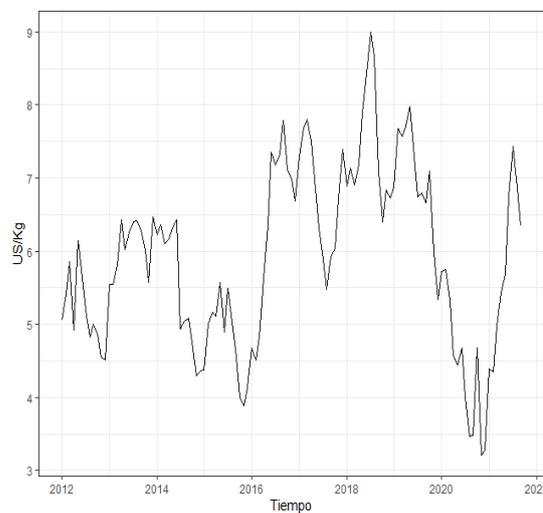
(a) Serie de Tiempo Salmón Salar



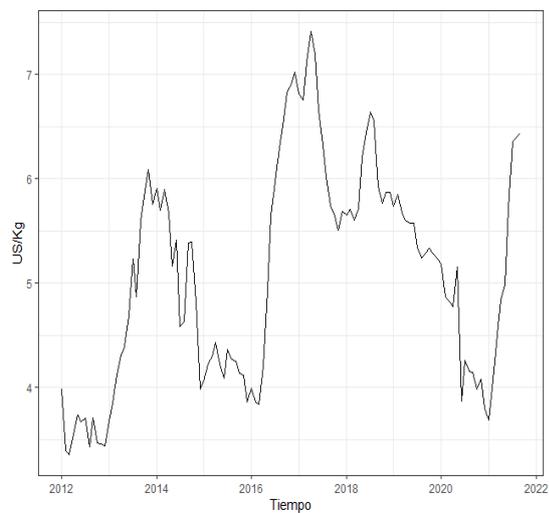
(b) Serie de Tiempo Salmón Salar USA



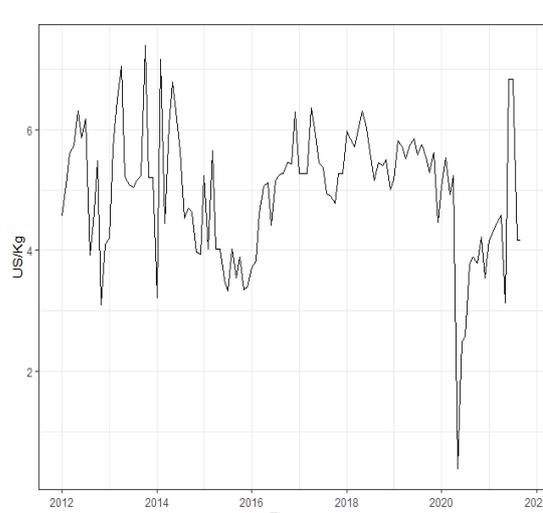
(c) Serie de Tiempo Salmón Salar Congelado USA



(d) Serie de Tiempo Salmón Salar Fresco USA



(e) Serie de Tiempo Salmón Salar Congelado Europa



(f) Serie de Tiempo Salmón Salar Fresco Europa

Figura 4.4: Series de Tiempo Salmón Salar

4.3. Modelos implementados

Esta tesis considera la implementación de modelos *Deep Learning* basados en *Recurrent Neural Networks* (RNN) y CNN, específicamente LSTM y ConvNet 1D respectivamente y su comparación con otros modelos predictivos, ampliamente usados para la predicción en series temporales como ARIMA, modelos clásicos de ML.

Los modelos clásicos de ML se implementan mediante las bibliotecas de código PyCaret y AutoKeras (descritos en subsección 4.3.1 y 4.3.2). Por otro lado, los modelos se implementan mediante la biblioteca *Keras*, la cual provee diferentes componentes y algoritmos de ML y DL. En función la figura 4.5 se muestra una representación del proceso llevado a cabo para la experimentación. El dataset empleado se divide en datos de prueba (20,0%) y datos de entrenamiento (80,0%) con el objetivo de evaluar su rendimiento mediante las métricas MAE, MSE y R^2 para cada modelo. Dividido el conjunto de datos, se inicia la etapa de preprocesamiento de los valores en cada serie de tiempo, en esta etapa se ejecuta la normalización de las series para reducir redundancia, finalizada a etapa se da inicio a la experimentación ejecutando cada una de las series temporales sobre modelos ARIMA, Enfoques Automáticos, modelos de ML y DL.

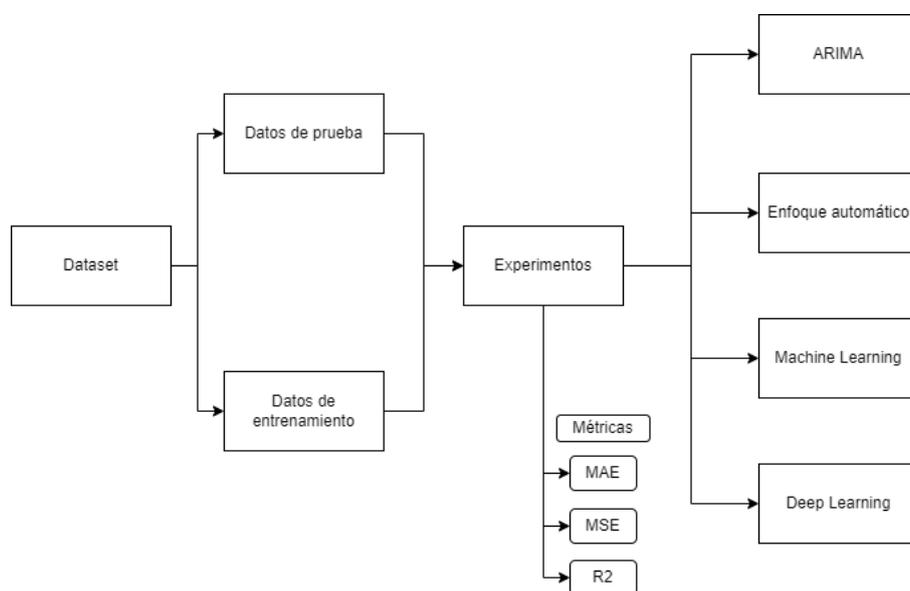


Figura 4.5: Diagrama experimentación implementada

El modelo LSTM como muestra la figura 4.6a se implementa, implementado mediante la clase LSTM de Keras. Los hiperparámetros considerados en cada capa LSTM son la cantidad de neuronas que definen la salida de la capa. En este experimento, el valor del dropout se aplica a los valores de entrada de la capa, y al mismo tiempo, el dropout recurrente es aplicado solo al estado recurrente. El resto de los hiperparámetros mantiene su valor por defecto. Cabe destacar que este modelo se aplica sobre las seis series de tiempo elaboradas (descritas en subsección 4.2.1). En cada capa oculta corresponde a una capa LSTM y para generar la salida se emplea una capa Dense, la cual permite reducir el vector de salida de la última capa.

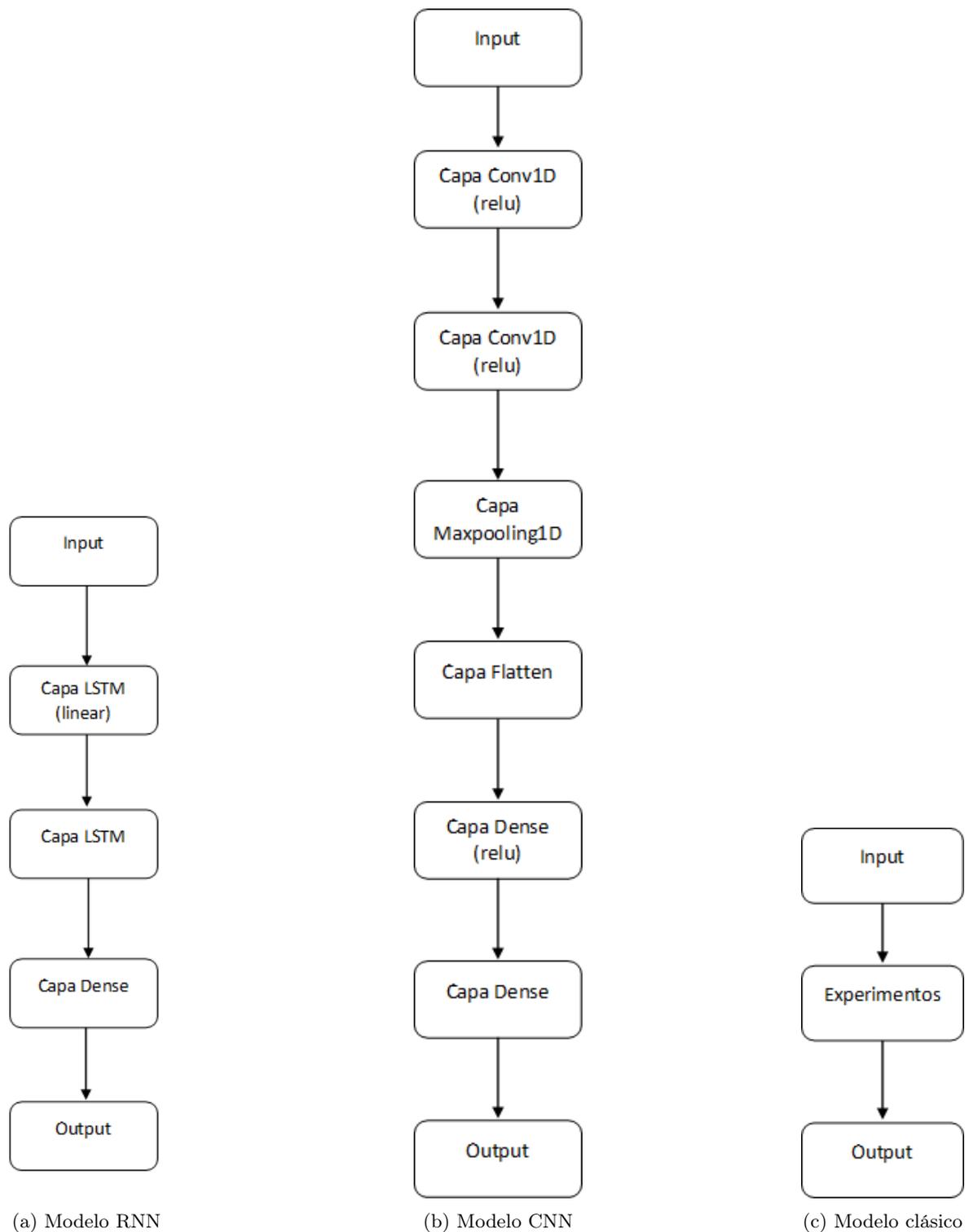


Figura 4.6: Esquemas de capas en implementación de cada modelo

Para el modelo CNN 1D figura 4.5b se emplea la clase Conv1D⁵. Los hiperparámetros utilizados en cada capa Conv1D son la cantidad de filtros, tamaño de kernel y función de activación, el resto de los hiperparámetros mantiene su valor por defecto. En la figura 4.6b se puede observar un esquema de la implementación para CNN. Se emplea una capa de entrada que recibe un conjunto de valores de las diferentes series de tiempo para realizar la estimación del valor precio. La capa de salida genera solo un valor que corresponde al valor del precio del salmón de acuerdo a las características que contienen cada

⁵<https://keras.io/api/layers/convolution1d/>

serie de tiempo utilizada. Cada capa oculta corresponde a una capa Conv1D. Previo a la capa de salida se incorpora una capa Flatten⁶, la cual aplanar el vector de datos que recibe de entrada. En la capa Dense se emplea una función relu, esta capa permite reducir el vector proveniente de la capa Flatten y producir un solo valor. Para los modelos clásicos (figura 4.5c) se utiliza ARIMA, modelos de ML y el Enfoque automático, utilizando las mismas series de tiempo para la ejecución de la experimentación, los valores de los hiperparámetros se mantienen por defecto en cada modelo utilizado.

4.3.1. Pycaret

Pycaret es una biblioteca de aprendizaje automático de código abierto y desarrollada en *Python*, que automatiza los flujos de trabajo de ML. Es una herramienta integral de gestión de modelos y ML que acelera el ciclo de experimentación y lo hace más productivo. En comparación con otras bibliotecas de ML de código abierto, realiza experimentos rápidos y eficientes, en sí, es un contenedor *Python* en torno a varias bibliotecas y marcos de ML, como *scikit-learn*, *XGBoost*, *LightGBM*, *CatBoost*, *spaCy*, *Optuna*, *Hyperopt*, *Ray*, entre otros⁷. Dentro de las características que tiene esta biblioteca es la preparación del dataset, entrenamiento del modelo, ajuste de hiperparámetros, visualización y selección del modelo. El uso de esta biblioteca de código busca generar de manera automatizada modelos de clásico de ML y medir su rendimiento mediante las métricas MAE, MSE y R^2 . La aplicación de los modelos implementados con esta biblioteca sobre cada una de las series de tiempo será bajo hiperparámetros por defecto, buscando encontrar tres principales modelos de clásicos de ML en cada una de las series de tiempo y evaluar su rendimiento. Esta experimentación no aplica el ajuste de los modelos encontrados.

4.3.2. AutoKeras

AutoKeras es un sistema de AutoML⁸ basado en *Keras*, desarrollado por DATA Lab en Texas A&M University. Un sistema de AutoML provee funciones que permiten la búsqueda automática de arquitecturas e hiperparámetros de modelos de DL⁹. Cuando se aplica a las redes neuronales, implica el descubrimiento de la arquitectura del modelo como los hiperparámetros utilizados para entrenar el modelo. Este tipo de biblioteca proporciona un enfoque simple para encontrar el modelo que se ajuste a cada una de las series de tiempo, buscando la optimización en cada uno de los hiperparámetros. Su evaluación se ajusta a las métricas antes señaladas y utilizadas en los experimentos anteriores.

La utilización de este tipo de herramientas será un punto de comparación con los diferentes modelos implementados y los propios ajustes de hiperparámetros que se llevarán a cabo durante los experimentos. En ese sentido, utilizar herramientas de ajuste manual de hiperparámetros puede simplificar las tareas de modificación de hiperparámetros.

⁶https://keras.io/api/layers/reshaping_layers/flatten/

⁷<https://pycaret.gitbook.io/docs/>

⁸Automated Machine Learning

⁹<https://autokeras.com/>

Capítulo 5

Resultados y Discusión

5.1. Diseño experimental

En esta sección se describe en detalle la etapa de entrenamiento de los modelos LSTM y CNN y su comparación con modelos *ARIMA*, modelos clásicos de ML para la estimación de precios en la industria del salmón. Los datos de entrada corresponden a diferentes series temporales descritas en la sección 4. En este experimento, la variable a estimar corresponde al precio (*price*) incorporando diferentes variables a considerar y que se encuentran detalladas en el cuadro 4.2.

La validación de los experimentos y los resultados obtenidos, es muy relevante la forma en que son empleados los datos y en este caso las series temporales. Es importante destacar que, medir el desempeño de los modelos sobre los conjuntos de datos empleados para datos de entrenamiento puede incurrir en errores de interpretación de los modelos. Es importante señalar que, del conjunto de datos inicial se reserva una fracción para realizar la prueba final, esto se utiliza al finalizar la experimentación. El conjunto de instancias restantes primero se emplea en realizar entrenamiento y selección de mejores valores para los hiperparámetros de cada modelo.

Esta tesis considera como un experimento, el entrenamiento del modelo y su evaluación para diferentes valores en un hiperparámetros específico, mientras que el resto de los hiperparámetros se mantienen sin variación en sus valores. Permitiendo establecer el efecto que tiene cada hiperparámetros de forma individual. Seleccionados los hiperparámetros y establecidos los escenarios iniciales, se da comienzo a la experimentación, estableciendo valores arbitrarios a cada hiperparámetros.

Para cada experimento es analizado solo un hiperparámetro, variando su valor y presentando los resultados en términos de las métricas MAE, MSE y R^2 . Estos resultados son presentados principalmente en cuadros para facilitar su interpretación.

Las estimaciones generadas por los modelos de DL serán contrastadas mediante un modelo paramétrico, que en este caso corresponde a los modelos *ARIMA*, con el fin de contrastar rendimiento para los modelos implementados. Adicionalmente a esto, es importante destacar que para se han incorporado una serie de variables que inciden en el precio del salmón según destino. Por tanto, es posible comparar el desempeño de los modelos de DL con el desempeño de los modelos *ARIMA* y establecer de forma clara comparaciones a nivel de métricas y disminución en el error de estimación del precio del salmón. Por tanto, para medir el error generado por las estimaciones, se presentará en este capítulo se presenta el valor de las métricas MAE, MSE y R^2 para los modelos entrenados como para los valores considerados como *baseline*. Estas métricas son las más representativas para diferentes problemas de estimación de precios o regresiones en ML y DL, esto se complementa con la revisión de la literatura presentada en la sección 3, siendo las más empleadas al momento de comparar estimaciones.

Cabe destacar que las métricas MAE y MSE son un promedio. Por otro lado, MAE destaca al calcular el valor absoluto de la diferencia entre la salida del modelo y el valor esperado, mientras que MSE calcula el cuadrado de esa diferencia. MAE se presenta como una métrica mas simple al generar un valor directo en el error de las estimaciones, pero oculta información al ser un promedio. En tanto, MSE emplea un exponente que amplifica errores de mayor envergadura, impidiendo que pasen desapercibidos al ser calculados como un promedio. Propiedades empleadas en su conjunto, al mismo tiempo, otras características que destacan las métricas MSE y MAE es que generan valores aplicados en conjuntos de valores esperados, esto, específicamente para una unidad de medida. En ese sentido, hace que los puntajes obtenidos por las métricas sean puntuales para un problema o conjunto de datos y su forma de interpretación no es sencilla. Respecto de R^2 , permite evaluar de forma lineal los valores de referencia

y las estimaciones generadas para los modelos implementados, cuantificando el grado de similitud entre una regresión lineal de un conjunto de datos y la línea de la función identidad $f(x) = x$, valores generados son encontrados entre -1 y 1, siendo 1, coincidencia entre las estimaciones y el valor de referencia, siendo bastante simple de interpretar. Es importante señalar que, R^2 viene a complementar las métricas MAE y MSE en su interpretación.

Respecto a lo mencionado, se considera como un experimento el entrenamiento del modelo y su evaluación para distintos valores de hiperparámetros y profundidad de la red neuronal (número de conexiones, cantidad de neuronas). Con esto, se buscará establecer el comportamiento que tiene cada modelo en función a los hiperparámetros seleccionados y tamaño de la red. Definida el tamaño de profundidad de la red neuronal, son seleccionados los hiperparámetros, se da comienzo a cada uno de los experimentos, estableciendo los escenarios iniciales, asignado diferentes valores en el tamaño de la red y de los hiperparámetros.

En cada experimento se analiza la profundidad de la red neuronal y el estudio de un hiperparámetro modificando su valor que puede adquirir, y se presentan los resultados en función a la métricas establecidas MAE, MSE y R^2 . Los resultados obtenidos se presentan mediante gráficos para facilitar su interpretación y al mismo tiempo visualización. Cada experimento presenta resultados gráficos de cada métrica. Adicionalmente, los resultados de los experimentos en cada uno de los modelos *ARIMA*, modelos clásicos de ML, LSTM y CNN.

5.2. Modelo *ARIMA*

Para el modelo *ARIMA* utilizado como *baseline*, se establecen parámetros por defecto, para establecer comparaciones entre modelos paramétricos y modelos de DL, comparando resultados mediante las métricas MAE, MSE y R^2 . La idea de utilizar este tipo de modelos es precisamente establecer el comportamiento de series sin patrones estacionales definidos y una alta variabilidad a variables externas que logran incidir en las estimaciones.

Cuadro 5.1: Resumen Métricas *ARIMA*

Serie de Tiempo	MAE	MSE	R2
Serie Salar	1,750	2,210	-
Serie Salar USA	2,130	2,600	0,580
Serie Salar Congelado USA	1,060	1,420	-
Serie Salar Fresco USA	1,530	1,780	0,560
Serie Salar Congelado Europa	0,940	1,270	-
Serie Salar Fresco Europa	0,860	1,060	0,150

De acuerdo a los resultados obtenidos en el cuadro 5.2, el modelo *ARIMA* su rendimiento de evalua mediante las métricas MAE y MSE y R^2 a fin de establecer parámetros comparativos con otros modelos. Si bien es cierto, los modelos paramétricos como *ARIMA* se ajustan bastante bien en series poco irregulares, este escenario cambia al momento de presentar series con distorsiones a través del tiempo y que no siguen un patrón definido. Entendiendo este tipo de modelos puede ser susceptible a la presencia de outliers, las series provenientes del salmón no presentan patrones estacionales de forma marcada, dependiendo de externalidades y de una serie de variables que inciden en su precio cada semana, mes y año. Si bien es cierto, este tipo de modelo es ampliamente utilizado en la predicción de precios.

5.3. Modelos clásicos de Machine Learning

Aprovechando la facilidad de implementación mediante Pycaret, se evalúan un alto número de modelos de ML clásicos para regresión, tales como *K Neighbors Regressor*, *Random Forest Regressor*, *Gradient Boosting Regressor* entre otros, para generar estimaciones sobre series de tiempo y cuantificar mediante las métricas MAE, MSE y R^2 el rendimiento de los modelos seleccionados en cada serie de tiempo. El cuadro 5.2 presenta los resultados de los 3 modelos clásicos de mejor rendimiento en función a las métricas MAE, MSE y R^2 , para cada serie de tiempo. Cabe mencionar que, no se abordaron etapas de ajustes de hiperparámetros en los modelos seleccionados.

Cuadro 5.2: Principales modelos con mejor rendimiento

Serie de Tiempo	Modelo	MAE	MSE	R2
Serie Salar Congelado USA	Random Forest Regressor	0,5834	0,5317	0,4603
	AdaBoost Regressor	0,6002	0,5371	0,4504
	Gradient Boosting Regressor	0,6141	0,5821	0,4210
Serie Salar Congelado Europa	AdaBoost Regressor	0,4102	0,2539	0,7338
	Extra Trees Regressor	0,4022	0,2928	0,7021
	Gradient Boosting Regressor	0,3907	0,2851	0,6842

En el cuadro 5.2 se observa que *K Neighbors Regressor* tiene el mejor rendimiento en las Serie Salar y Serie Salar USA. Por otro lado, la Serie Salar Congelado USA tiene como principal modelo *Random Forest Regressor*. Respecto de la Serie Salar Congelado Europa *AdaBoost Regressor* obtuvo las mejores métricas de evaluación al ejecutar el proceso de selección automática. Finalmente, Serie Salar Fresco Europa presentó en *Random Forest Regressor* el modelo con el mejor rendimiento. La tabla con los resultados ejecutados sobre la completitud de las series de tiempo, pueden ser revisados en Anexo 1.

5.4. Modelos Deep Learning con ajuste automático

Mediante la biblioteca AutoKeras se ejecutó una serie de experimentos para evaluar cada una de las series de tiempo, teniendo como métricas de evaluación MAE, MSE y R^2 , el cuadro 5.3 presenta los resultados obtenidos.

Cuadro 5.3: Resumen Métricas Autokeras

Serie de Tiempo	MAE	MSE	R2
Serie Salar	0,826	1,160	0,045
Serie Salar USA	0,909	1,529	0,312
Serie Salar Congelado USA	0,737	0,851	0,409
Serie Salar Fresco USA	1,021	1,496	0,101
Serie Salar Congelado Europa	0,401	0,234	0,809
Serie Salar Fresco Europa	0,551	0,489	0,544

De acuerdo a la información presente en el cuadro anterior, el ajuste de hiperparámetros automatizado, presenta rendimientos deficientes para ejecutar estimaciones en cada una de las series de tiempo. Por otro lado, en cada ejecución MAE y MSE comienzan a disminuir.

5.5. Long Short-Term Memory

Para este tipo de modelo que se desprende de las RNN, los valores iniciales de los hiperparámetros son:

Modelo LSTM Inicial

- Neuronas:100
- Capas ocultas: 2
- Función de activación: linear
- Dropout recurrente: 0.0
- Dropout: 0.0
- Optimizador: adam

Los conjuntos de datos empleados en cada uno de los experimentos sobre un modelo LSTM se detallan a continuación:

- Serie Salmón Salar.
- Serie Salmón Salar USA.
- Serie Salmón Salar Congelado USA.
- Serie Salmón Salar Fresco USA.
- Serie Salmón Salar Congelado Europa.
- Serie Salmón Salar Fresco Europa.

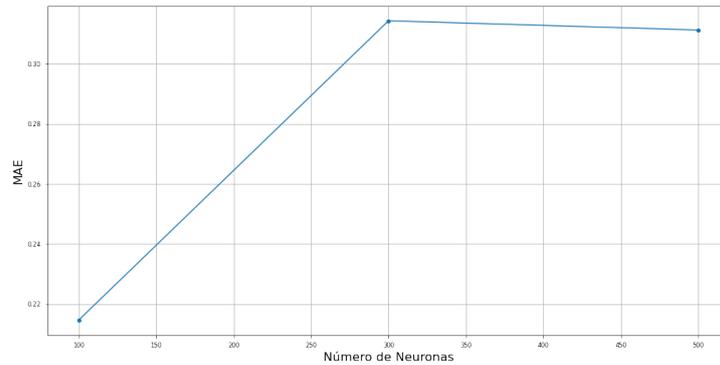
Es importante destacar que, para efectos de la experimentación inicial contiene a la Serie Salmón Salar en forma completa, y para los siguientes experimentos, se evaluaron los hiperparámetros función de activación, dropout y optimizador.

Para las series antes mencionadas contarán con mismo porcentaje de entrenamiento, que en este caso será un 80,0% de las instancias y un 20,0% de las instancias para validación.

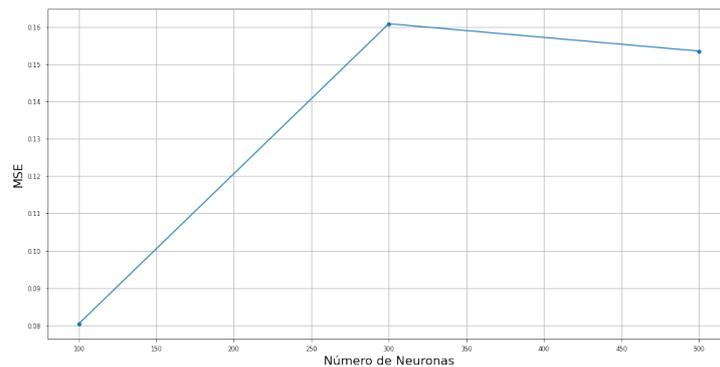
5.5.1. Serie Salmón Salar

Neuronas

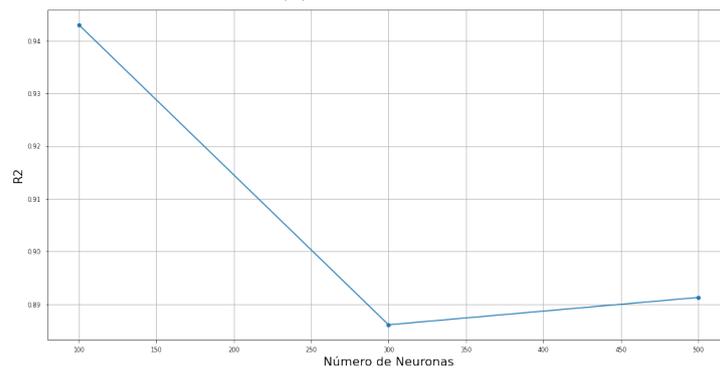
Los experimentos corresponden a la evaluación del rendimiento en función a la cantidad de neuronas para las métricas MAE, MSE y R^2 disponibles en la figura 5.1. El modelo se entrena con hiperparámetros iniciales y diferentes cantidades de neuronas por capa. El modelo se entrena con 100, 300 y 500 neuronas por capa. De las gráficas obtenidas, las métricas presentan los mejores rendimientos en 100 neuronas, y a medida que la cantidad de neuronas aumenta, el rendimiento para MAE, MSE y R^2 comienza a disminuir.



(a) Métrica MAE



(b) Métrica MSE

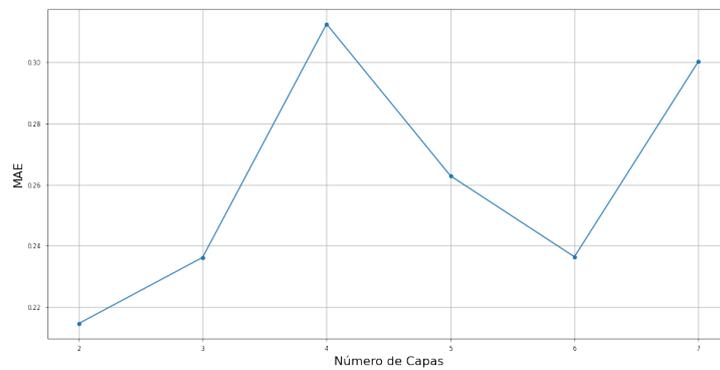


(c) Métrica R^2

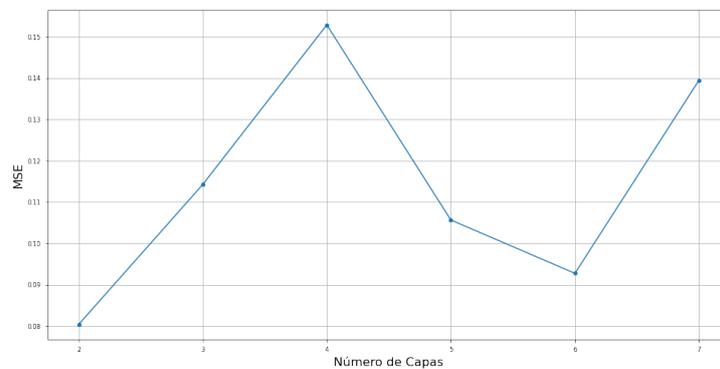
Figura 5.1: Métricas N° de Neuronas Serie de Tiempo Salmón Salar.

Capas

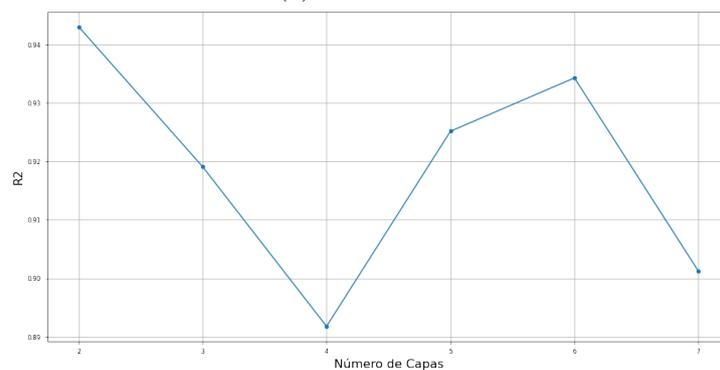
El siguiente experimento corresponde a la evaluación del rendimiento en función a la cantidad de capas, se realizan pruebas con 2, 3, 4, 5, 6 y 7 capas. Los resultados evalúan el rendimiento del modelo de acuerdo a las métricas MAE, MSE y R^2 y se encuentran disponible en la figura 5.2. Considerando el mejor resultado del experimento anterior, en este experimento se emplean 100 neuronas por capa y el resto de hiperparámetros iniciales para el conjunto de datos.



(a) Métrica MAE



(b) Métrica MSE

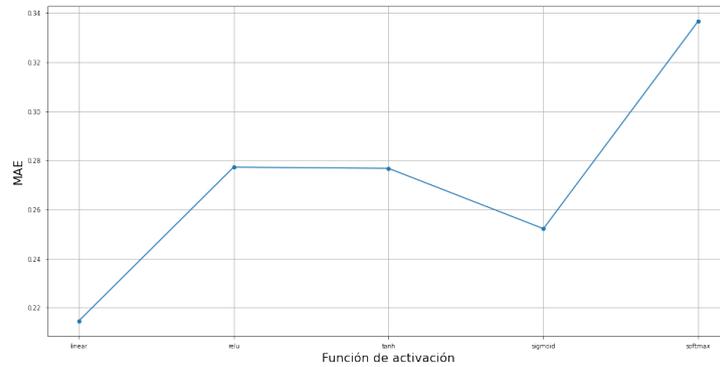


(c) Métrica R^2

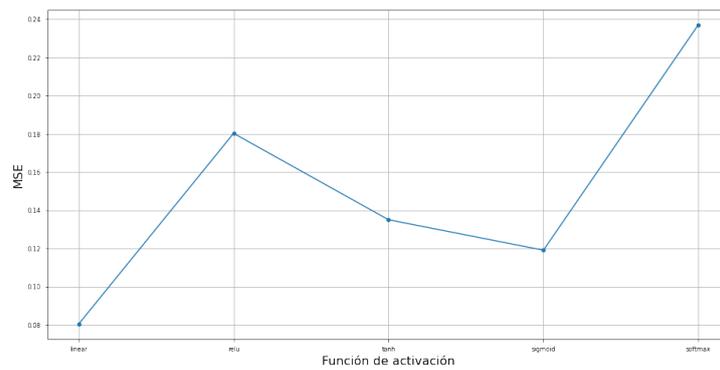
Figura 5.2: Métricas N° de Capas Serie de Tiempo Salmón Salar.

Función de activación

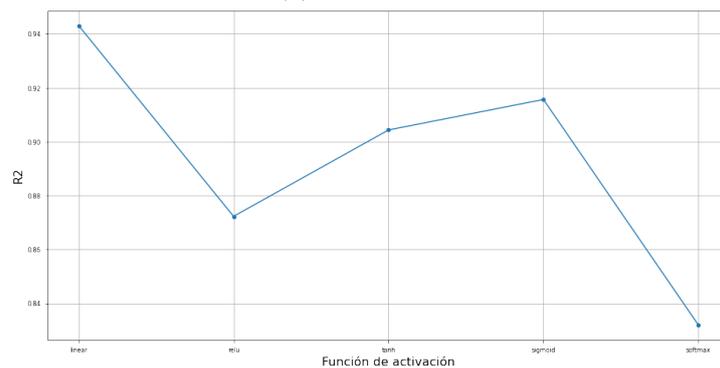
El tercer experimento corresponde a la evaluación del rendimiento respecto a la función de activación en las neuronas, esto, de acuerdo a las métricas MAE, MSE y R^2 , resultados disponible en la figura 5.3. De acuerdo con los resultados de los experimentos anteriores, en este experimento se emplean 2 capas con 100 neuronas, al mismo tiempo, los hiperparámetros mantienen valores iniciales detallados anteriormente. Las pruebas consisten en la funciones de activación linear, relu, tanh, sigmoid y softmax, (descrita en sección 2.6). Para las métricas MAE, MSE y R^2 se opbservan mejores rendimientos con la función de activación linear a diferencia de las demás funciones de activación.



(a) Métrica MAE



(b) Métrica MSE

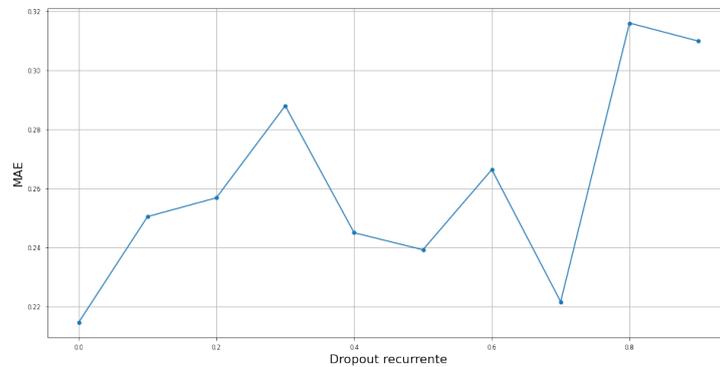


(c) Métrica R^2

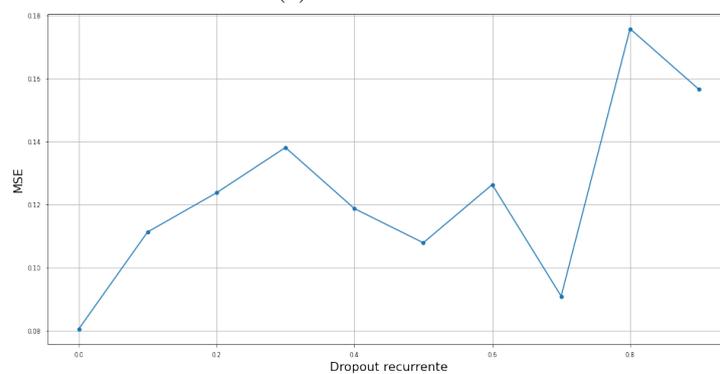
Figura 5.3: Métricas Función de Activación Serie de Tiempo Salmón Salar.

Dropout recurrente

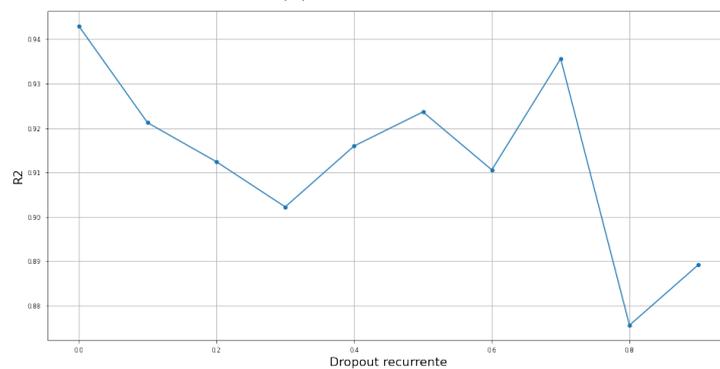
El cuarto experimento corresponde a evaluar el rendimiento para distintos valores de dropout recurrente, se evalúan las métricas MAE, MSE y R^2 , los resultados se presentan en la figura 5.4. Los valores utilizados corresponden a 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8 y 0.9 para las pruebas ejecutadas. Según las gráficas el mejor rendimiento es sin dropout recurrente.



(a) Métrica MAE



(b) Métrica MSE

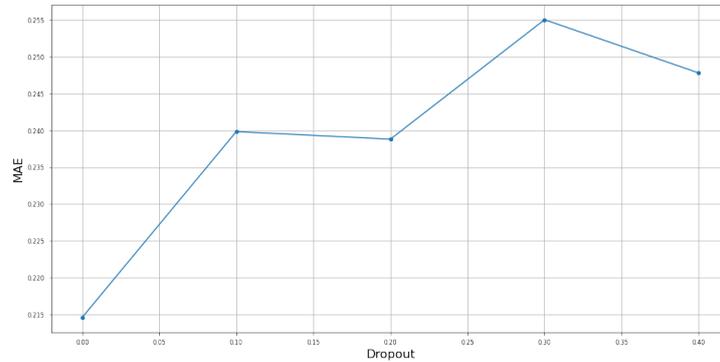


(c) Métrica R^2

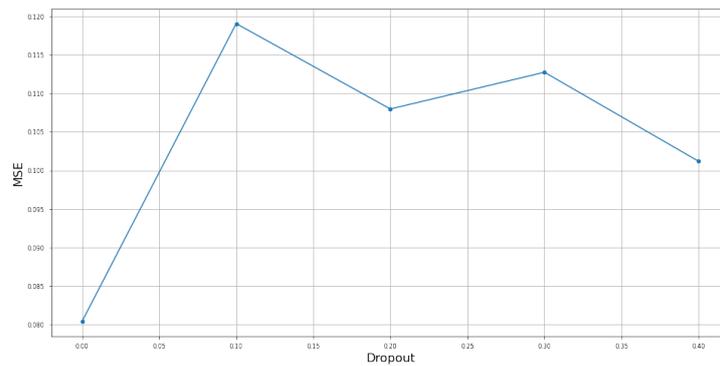
Figura 5.4: Métricas Dropout Recurrente Serie de Tiempo Salmón Salar.

Dropout

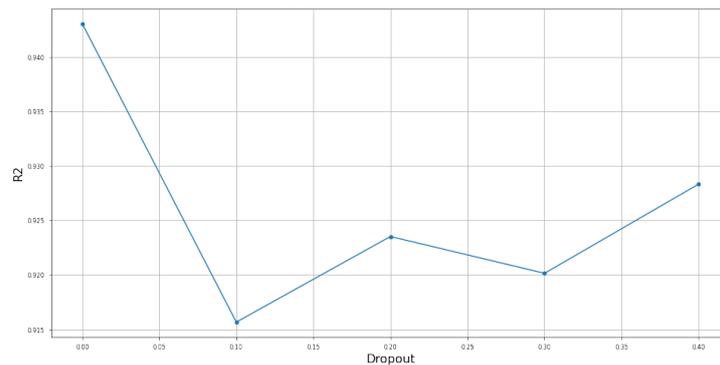
El cuarto experimento corresponde a la evaluación del rendimiento para distintos valores en las capas dropout, resultados evaluados para las métricas MAE, MSE y R^2 disponibles en la figura 5.5. Se utilizaron valores 0.1, 0.2, 0.3 y 0.4 para cada una de las pruebas. Según figura se probó sin dropout (0) también. Y el mejor resultado es sin dropout. El mejor rendimiento se obtuvo con un dropout 0.1 para MAE y 0.2 para MSE. Por otro lado, R^2 obtiene un mejor rendimiento para un dropout de 0.2. De acuerdo con resultados de experimentos anteriores con 2 capas ocultas de 100 neuronas y la función de activación lineal y sin dropout recurrente.



(a) Métrica MAE



(b) Métrica MSE

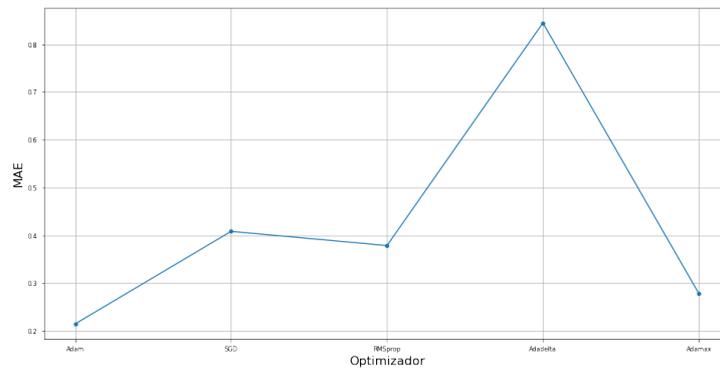


(c) Métrica R^2

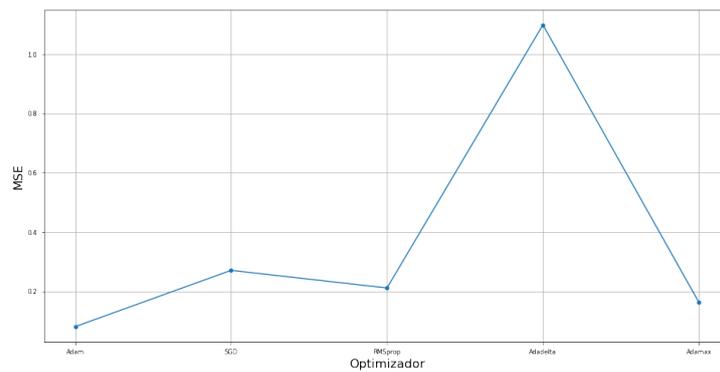
Figura 5.5: Métricas Dropout Serie de Tiempo Salmón Salar.

Optimizador

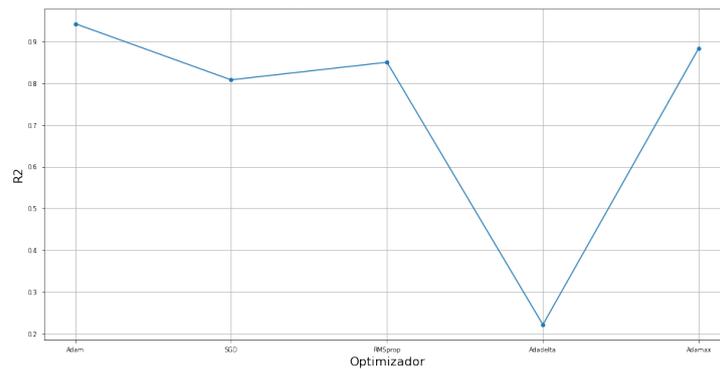
Finalmente, el quinto experimento para el modelo LSTM y cada una de las series temporales corresponde a la evaluación de los optimizadores con el objetivo de medir el rendimiento, en este caso se evalúa Adam, SGD, RMSprop, Adadelta y Adamax (descritos en sección 2.5). Los resultados para cada una de las métricas MAE, MSE y R^2 se encuentran disponibles en la siguiente figura 5.6. De acuerdo con los experimentos anteriores, con 2 capas ocultas de 100 neuronas cada una, la función de activación Adam y sin dropout ni dropout recurrente, el mejor optimizador en este caso resulta ser Adam. Los resultados obtenidos se encuentran disponibles en Anexo N°1.



(a) Métrica MAE



(b) Métrica MSE



(c) Métrica R^2

Figura 5.6: Métricas Optimizador Serie de Tiempo Salmón Salar.

5.5.2. Análisis otras series temporales

El siguiente apartado presenta los experimentos realizados sobre el conjunto de series temporales elaboradas, las métricas de evaluación corresponden a MAE, MSE y R^2 y el modelo inicial utilizando en la experimentación el conjunto de series temporales elaboradas detalla a continuación:

Modelo LSTM inicial

- Neuronas:100
- Capas ocultas: 2
- Función de activación: linear
- Dropout: 0.0
- Optimizador: adam

Cabe destacar que la experimentación realizada sobre cada serie de tiempo contempla se mantiene fijo el hiperparámetro número de neuronas (100 neuronas) y la cantidad de capas ocultas (dos capas). Esta decisión se basó en pruebas preliminares que mostraron, para las diferentes series, que al aumentar el número de neuronas o capas ocultas, el error de predicción aumenta.

Función de activación

El primer experimento ejecutado sobre las series temporales se basa en la modificación como hiperparámetro de la función de activación, que para este caso, se utilizan las funciones linear, relu, tanh, sigmoid y softmax y las métricas utilizadas MAE, MSE y R^2 .

Los resultados obtenidos destacan la función de activación linear con las mejores métricas tanto para MAE (0,267), MSE (0,159) y R^2 (0,917) para la Serie Salar USA. Respecto de la Serie Salar Congelado USA, presenta los mejores rendimientos utilizando como función de activación linear MAE (0,363), MSE (0,278) y R^2 (0,751). Por otro lado, la Serie Salar Fresco USA tiene resultados en las métricas MAE (0,409), MSE (0,246) y R^2 (0,829) utilizando la función de activación sigmoid. La Serie Salar Congelado Europa, tiene métricas MAE (0,241), MSE (0,105) y R^2 (0,838) utilizando sigmoid como función de activación. Finalmente, el último experimento realizado fue sobre la Serie Salar Fresco Europa presenta utilizando softmax presenta un MAE (0,787), por otro lado, los mejores rendimientos en MSE (1,500) y R^2 (0,168) se obtienen mediante la función de activación sigmoid. El resumen de los resultados obtenidos sobre la experimentación ejecutada sobre la función de activación de detallan en el cuadro 6.4 a modo de resumen disponibles en el Anexo N°1.

Dropout

El segundo experimento ejecutado sobre cada una de las series de tiempo corresponde a dropout con valores de 0.1, 0.2, 0.3 y 0.4, evaluando las métricas MAE, MSE y R^2 . Los resultados obtenidos se presentan en el Anexo N°1, cuadro 6.5. Los resultados muestran que, en general para todas las series, el desempeño disminuye a medida que la tasa de dropout aumenta.

La experimentación ejecutada sobre la Serie Salar USA tuvo como principales resultados una MAE (0,270) y MSE (0,169) para un valor de dropout igual a 0.4, respecto de R^2 (0,913) este se obtiene con un dropout de 0.4. La ejecución efectuada sobre la Serie Salar Congelado USA, presentó los principales rendimientos con dropout igual a cero tanto para MAE, MSE y R^2 , que presentaron resultados de 0,363, 0,278 y 0,751 respectivamente. La Serie Salar Fresco USA, al igual que la serie de tiempo analizada anteriormente, los mejores rendimientos se obtienen con dropout igual a cero, tanto para MAE (0,409), MSE (0,246) y R^2 (0,829). Respecto Salar Congelado Europa un dropout de 0.3 presenta los mejores rendimientos para cada una de las métricas MAE, MSE y R^2 , presentando valores de 0,209, 0,081 y 0,875 respectivamente. Finalmente, la última evaluación corresponde a Serie Salar Fresco Europa tiene valores MAE (0,832), MSE (1,500) y R^2 (0,168) con dropout igual a cero.

Cabe destacar que en el caso de la serie Salar Fresco Europa los valores de error son mayores, resultado debido a su menor tamaño (117 instancias), lo que dificulta el adecuado entrenamiento del modelo.

Optimizador

Finalmente, la última experimentación realizada corresponde a evaluar MAE, MSE y R^2 , en cada optimizador. Los optimizadores utilizados en este experimento corresponden a Adam, SGD, RMSprop, Adadelta y Adamax. El resumen de los resultados obtenidos se presentan en el cuadro 6.6 disponible en el Anexo N°1.

El experimento ejecutado sobre la serie de tiempo Salar USA tuvo en el optimizador Adam el mejor resultado sobre la métrica MAE (0,267), por otro lado, con RMSprop se presentaron los mejores rendimientos en MSE (0,141) y R^2 (0,927). En la Serie Congelado USA se presentaron los mejores rendimientos en MAE (0,363), MSE (0,278) y R^2 (0,751) fueron obtenidos mediante el optimizador Adam. Respecto de la serie de tiempo Salar Fresco USA, se presentaron los mejores rendimientos para el modelo ejecutado utilizando el mismo optimizador antes mencionado. Otra experimentación ejecutada fue sobre la Serie Salar Congelado Europa y de acuerdo a lo presentado en el cuadro 5.7, se tiene que MAE (0,209), MSE (0,081) y R^2 (0,875) fueron los principales resultados obtenidos utilizando como optimizador Adam. Finalmente, la serie de temporal Salar Fresco Europa, se puede concluir que los principales rendimientos para MAE (0,789), MSE (1,246) y R^2 (0,309), se consiguen utilizando como optimizador RMSprop.

Adicionalmente, es importante señalar que a medida que las series de tiempo contienen menor información, o los períodos analizados son más acotados, los modelos evaluados van perdiendo eficacia al momento de generar estimaciones impactando de forma negativa en cada una de las métricas utilizadas a fin de generar una evaluación del modelo.

5.6. Convolutional Neural Networks

El siguiente experimento consta de evaluar un modelo CNN sobre las serie de tiempo me mejor y peor rendimiento sobre las métricas MAE, MSE y R^2 y poder establecer un análisis comparativo de sus hiperparámetros.

El modelo de CNN (1D) se realizó considerando los siguientes valores de hiperparámetros:

- Filtros: 64
- Tamaño kernel: 3
- Capas: 6
- Función de activación: relu
- Optimizador: Adam

El conjunto de datos empleados para este experimentos considera las Series de Tiempo Salmón Salar y Salmón Salar Fresco Europa. La selección de estas series de tiempo radica en el comportamientos que obtuvieron de las métricas MAE, MSE y R^2 en experimentos anteriores. Por una lado, la serie temporal Salmón Salar presenta el mejor rendimientos sobre las métricas antes mencionadas utilizando un modelo LSTM. Por otro lado, el menor desempeño de los modelos evaluados se obtuvo con la serie Salmón Salar Fresco Europa, el cuadro 5.8 presenta los resultados obtenidos.

Cuadro 5.4: Resumen Métricas CNN

Serie de Tiempo	MAE	MSE	R2
Serie Salar	0,137	0,027	0,981
Serie Salar Fresco Europa	0,272	0,211	0,883

Como se puede observar a partir del cuadro anterior, la Serie Salmón Salar presentan rendimientos bastante eficientes para las métricas MAE, MSE y R^2 , misma situación ocurre con la Serie Salmón Salar Fresco Europa. En los siguientes apartados se presentará un análisis comparativos entre modelos de ML y DL.

5.7. Análisis comparativos

En este apartado se presenta un análisis comparativo sobre los experimentos ejecutados con anterioridad, de acuerdo a las métricas MAE, MSE y R^2 , identificando qué modelos presentan mejores rendimientos para la generación de estimaciones, en este caso se comparan modelos paramétricos como ARIMA, modelos de ML y modelos de DL como lo son LSTM y CNN.

Cuadro 5.5: Principales modelos y resultados

Modelo	Serie de Tiempo	MAE	MSE	R2
ARIMA	Serie Salar USA	2,130	2,600	0,580
K Neighbors Regressor	Serie Salar	0,849	1,445	0,391
Deep Learning (AutoKeras)	Serie Salar Congelado Europa	0,401	0,234	0,809
LSTM	Serie Salar	0,215	0,080	0,943
CNN 1D	Serie Salar	0,137	0,027	0,981

Como se pudo observar en el cuadro 5.9, la metodología empleada muestra que modelos de *Deep Learning* como LSTM y CNN 1D efectivamente mejoran el desempeño de las estimaciones de precios de salmón, en función a los datos empleados. Señalar que, la Serie Salar se presenta con los principales rendimientos a nivel de métricas, esto obedece principalmente a la longitud de esta serie de tiempo, por tanto, los modelos LSTM y CNN 1D pueden generar estimaciones con mayor robustez.

5.7.1. Sobre optimización de hiperparámetros

Una primera instancia en la experimentación radica en evaluar modelos de Machine Learning y Deep Learning con de ajuste automático de hiperparámetros ejecutados sobre cada una de las series temporales. La ejecución y búsqueda de modelos clásicos de ML buscó establecer un punto de comparación con respecto a modelos basados en DL como también establecer a nivel de métricas MAE, MSE y R^2 el rendimientos de de los modelos seleccionados de forma automática. En este caso destacan los modelos *K Neighbors Regressor* en la Serie Salar, *Gradient Boosting Regressor* para Serie Salar USA. En el caso de la Serie Salar Congelado USA el modelo *Random Forest Regressor* presenta el mejor rendimiento. Por otro lado, Serie Salar Fresco USA destacó *Extra Trees Regressor*. En tanto, *AdaBoost Regressor* se poció con el mejor rendimiento en la Serie Salar Congelado Europa. Finalmente, el modelo *Random Forest Regressor* tuvo el mejor rendimiento en la Serie Salar Fresco Europa, pero a diferencia de otras series de tiempo, presenta métricas MAE, MSE y R^2 por debajo a lo presentado en los experimentos anteriores.

La evaluación de modelos DL con ajuste automático de hiperparámetros permitió evaluar el comportamiento que tienen este tipo de de enfoques sobre las series temporales elaboradas. Una de las características importante que se logro determinar es que en cada ejecución, los rendimientos de los modelos mejoran pero en relación a la ejecución anteriores métricas obtenidas mediante modelos ML y modelos DL con ajuste automático de hiperparámetros. En ese sentido, los rendimientos obtenidos en cada serie de tiempo no presenta resultados óptimos al momento de generar estimaciones.

La evaluación de hiperparámetros permitió estudiar detalladamente ajustes según la serie de tiempo a evaluar, en este caso para el modelo LSTM se considera numero de neuronas, cantidad de capas ocultas, función de activación, la inclusión de capas Dropout y Dropout Recurrente, como también optimizador, esto para evaluar la Serie Salmón Salar, respecto de otras series de tiempo, se fueron tomados en cuenta función de activación, Dropout y optimizador. Para la Serie Salmón Salar, se observó que a medida que aumenta el número de neuronas, presenta un rendimiento decreciente. Los resultados presentados por las capas ocultas, los rendimientos no presentan una tendencia clara, al aumentar el numero de capas. Por otro lado, la función de activación de peor rendimiento es Softmax, al mismo tiempo, linear presenta los mejores rendimientos a lo largo de la experimentación. Respecto de la utilización de Dropout recurrente no genera cambios importantes en el modelo, misma situación ocurre al momento de ejecutar Dropout, el modelo no incrementa su rendimiento al generar cambios en el Dropout. Para el análisis de los optimizadores presentan resultados más consistentes, esto exceptuando Adadelta, que presentó el menor rendimiento a nivel de métrica. En general se puede observar resultados disimiles de acuerdo a los experimentos realizados, esto, en directa relación al comportamiento que presenta la serie de tiempo, sin patrones definidos y afectada por ciertos factores externos que logran incidir en su evolución a través del tiempo.

Respecto de otras series de tiempo, y como se mencionó anteriormente, la evaluación de hiperparámetros considero función de activación, Dropout y optimizador. En general, los resultados obtenidos presentan una misma tendencia, en función a la experimentación realizada. Es necesario considerar que este tipo de series temporales presentan diferentes tamaño, y da cuenta de como los modelos LSTM pueden lograr estimar precios de forma estable, en función al tipo de arquitectura empleada y por otro lado, al tamaño del modelo. Cabe destacar que, las Series Salmón Salar Congelado y Fresco con destino a Europa tiene métricas pueden presentar ciertas discrepancias respecto de otras series de tiempo, debido debido a que presentan menos información.

Para la evaluación del modelo CNN se consideraron la cantidad de filtros, tamaño de kernel, capas, función de activación en capas ocultas y optimizador. Cabe señalar que la experimentación mediante CNN utilizó tanto la serie de tiempo con el mejor rendimiento de los experimentos anteriores, como la serie con el menor rendimiento. En este caso, la ejecución utilizó la Serie Salar y Serie Salar Fresco Europa. Los resultados obtenidos en función a los hiperparámetros definidos dan cuenta que un modelo CNN presenta un rendimiento superior por sobre modelos de ML, como también por sobre modelos LSTM. Los resultados obtenidos muestran que este tipo de modelo claramente tiene mejores resultados, independiente del tamaño de la serie y las variables que inciden en su variabilidad a través del tiempo, logrando generar estimaciones con bajo error.

Para definir el cambio en el valor de los hiperparámetros se realizó mediante ajuste automático, donde la propia arquitectura selecciono de acuerdo a los rendimientos obtenidos en MAE, MSE y R^2 . Este punto es bastante importante, ya que la selección automática de hiperparámetros y modelos de ML pueden incurrir en sesgos que finalmente impactan a nivel de métricas y por sobre todo, al momento de generar estimaciones bastante erráticas. Por otro lado, la serie de mejor desempeño en la experimentación (Serie Salar) presentó para un modelo LSTM una disminución en su rendimiento entre el primer experimento y el

último en las métricas MAE, MSE y R^2 esto en un -38,7%, -68,7% y -19,3% en promedio, impactando directamente en las estimaciones. Ahora, la comparación entre las métricas MAE, MSE y R^2 con el mejor modelo LSTM, se obtiene una mejora de 56,6%, 197,8% y 4,0% en el rendimiento del modelo CNN implementado.

Cada experimento se realizó de forma consecutiva, el orden de modificación de hiperparámetros fue arbitrario, en ese sentido, el mejor valor obtenido para un hiperparámetro se emplea en el siguiente experimento. Considerando esta característica de la experimentación, permite establecer que los valores de los hiperparámetros que resultan de la experimentación, son en consecuencia del orden en que se evaluaron los hiperparámetros. Por ejemplo, en el caso del modelo LSTM, primero se evalúa el número de neuronas, y donde se genera el mejor desempeño corresponde a 100 neuronas, lo que condiciona los resultados del experimento en el número de capas. Esto tiene efecto directo en los experimentos siguientes y para los hiperparámetros posteriores.

Capítulo 6

Conclusiones y Trabajos Futuros

En esta tesis se busca comparar ARIMA, modelos clásicos de ML como también con ajuste automático para la predicción de precios de exportación de salmón y especialmente, el aporte de modelos de DL en reducir el error de las estimaciones. Se utilizó un conjunto de datos disponible en el SNA, que presenta información sobre de las exportaciones nacionales, con sus respectivos destinos. A partir de estos datos, se elaboraron diferentes series temporales que luego fueron utilizadas como entrada para los modelos a evaluar. Para lograr el objetivo principal de la tesis que es, evaluar modelos de Deep Learning (DL) sobre datos de exportaciones nacionales provenientes desde la salmonicultura en Chile, con el fin de medir mejoras en el desempeño con respecto a modelos clásicos en la estimación de precios de salmón y se propusieron tres objetivos resumidos en (1) selección de modelos de DL, (2) su implementación y (3) evaluación. La creación de distintas series temporales se realizó mediante un procesamiento exhaustivo, que incluyó exclusión de variables que no aportaban al desarrollo de un modelo predictivo, detección de códigos arancelarios que identificaban a productos derivados del salmón, revisión y eliminación de instancias con valores ausentes, tratamiento de variables, normalización de las variables, empleando promedio y desviación estándar. Esto permitió la generación de seis series temporales y la incorporación de variables económicas que buscaban interpretar el comportamiento del precio de exportación.

El objetivo específico 1 (OE1) señala: *Seleccionar modelos predictivos propuestos en la literatura que permitan realizar estimaciones sobre precios de salmón*. Su completitud se verifica mediante los resultados expuestos en la Sección 2 de conceptos preliminares y Sección 3, del estado del arte, donde se presenta aquellos trabajos y estudios realizados del análisis de la literatura. El desarrollo y construcción del estado del arte requirió la revisión de diferentes trabajos provenientes de revistas científicas y artículos, permitiendo identificar modelos basados en *Deep Learning* (Long Short-Term Memory Recurrent Neural Network y 1D Convolutional Neural Network) utilizados en la estimación de precios para diferentes sectores económicos. Sin embargo, cabe destacar que no se encontraron trabajos de DL enfocados en la estimación de precios para el mercado del salmón a nivel nacional e internacional. El estudio y definición de los algoritmos se extendió hacia el estudio de otras fuentes de información como libros del área, esto permitió identificar modelos de características similares a los propuestos en el estado del arte.

La selección de modelos de *Deep Learning* fue realizada mediante el estudio y revisión de la literatura antes mencionada. Como primera instancia, no se encontraron trabajos específicamente en el área de la industria del salmón, se lograron identificar trabajos que utilizaban técnicas clásicas en la predicción de precios (ARIMA, VAR, entre otros) y modelos tradicionales de *Machine Learning*. Debido a esto, fueron considerados trabajos que utilizaban técnicas de DL para estimar precios, en distintas áreas y sectores económicos.

Se revisaron de forma exhaustiva 25 trabajos que emplean algún método de ML y DL para la predicción de precios. Solo [3] destaca utilizando técnicas de ML en la predicción de precios del salmón, el resto de los modelos utilizados consisten en técnicas parámétricas. Por otro lado, se estableció ampliar la búsqueda a trabajos de DL y esta segunda revisión contempló revisar 25 trabajos que empleaban alguna técnica de DL para estimar precios, destacan la utilización de modelos LSTM y CNN para implementar modelos predictivos y en este caso, se utilizó como base el desarrollo e implementación de una red LSTM y CNN 1D para generar predicciones sobre las series temporales.

En general, de la revisión de la literatura se puede concluir que (1) no se encontraron trabajos que evalúan modelos de DL para el mercado del salmón, no así técnicas tradicionales de ML. Esto puede ser consistente con la dificultad de obtener información y la complejidad de su manejo para implementar estos modelos. En el caso de esta tesis, se emplean datos públicos provistos por el SNA, los cuales deben

ser tratados adecuadamente antes de ser utilizados como entrada por los modelos. Se consideran otras fuentes públicas de información tales como IPC, tipo de cambio, valor del petróleo a nivel internacional, para robustecer el modelo, considerando variables económicas y que puedan explicar el comportamiento del precio a nivel nacional e internacional. (2) En los trabajos no se encontró un esquema que facilite la comparación de resultados de diferentes trabajos y que pueda permitir llevar un seguimiento en la estimación de precios en el área de estudio. Es sumamente importante establecer etapas claras en el preprocesamiento del conjunto de datos, como también definir codificaciones que logren identificar productos salmónidos provenientes de registros administrativos del SNA. Al mismo tiempo, validez de las técnicas, definir métricas para analizar los datos. Finalmente, un esquema de modelamiento de los datos, permitiría tener claridad en como los modelos de DL pueden interactuar con las series de tiempo.

El objetivo específico 2 (OE2) señala: *Implementar algoritmos seleccionados basados en DL y contrastar estimaciones con algoritmos tradicionales*. Su completitud se verifica mediante los resultados de la preparación y selección de los datos, presentados en la Sección 4 y los resultados del diseño e implementación de la metodología. Implementar los diferentes modelos, requirió la correcta preparación y formato de los datos para hacerlos compatibles con los modelos. Se debió establecer una metodología clara que logre identificar desde el conjunto de datos, los productos provenientes desde las exportaciones de salmón, con el fin de desarrollar de forma adecuada la realización de los experimentos y definición de los hiperparámetros de los diferentes modelos, considerando sobreentrenamiento, generalización, validación, etc.

La implementación de los modelos de *Recurrent Neural Networks* y *Convolutional Neural Networks* se realiza mediante la biblioteca de código abierto *Keras*, que dispone de APIs que facilitan la construcción de arquitecturas y la modificación de sus hiperparámetros. Para implementar los modelos de DL se aplican técnicas de preprocesamiento al conjunto de datos original de 99.353 instancias, del que se desprenden seis series temporales con diferentes características y longitud de información. Los conjuntos de datos desarrollados se experimentan bajo los modelos LSTM y CNN 1D, los datos se preparan para disponer de la variable a estimar de cada conjunto de datos y que sirve de entrada para los modelos. Estas variable de entrada consisten en la variable precio (US/Kg) presente en cada serie de tiempo, adicionalmente se incorporan atributos adicionales para la estimación de precios. Las técnicas de tratamientos de los datos como la normalización están basadas en técnicas de los trabajos seleccionados de la revisión de la literatura, mencionadas en las Secciones 3 y 4.

El objetivo específico 3 (OE3) señala: *Evaluar el desempeño de los distintos algoritmos implementados según métricas y pruebas estadísticas*. Su completitud se verifica mediante los resultados presentados en la Sección 5, donde se señala el desempeño de los modelos de acuerdo a las métricas MAE, MSE y R^2 . En la Sección 5.10, se presentan resultados de las pruebas finales, donde se considera la comparación de métricas con modelos tradicionales en el modelamiento de series temporales y la comparación entre modelos clásicos de ML y librerías automáticas que permiten ajustar hiperparámetros. Finalmente en la Sección 5 se presenta el desempeño de los modelos.

Cabe destacar que, la evaluación de los modelos se establece en dos etapas (1) Ajuste y optimización de hiperparámetros y (2) prueba final, con la cual se establecen comparaciones con los diferentes modelos evaluados.

Finalmente y en función a los objetivos planteados, se puede concluir que, los modelos de DL implementados LSTM y CNN 1D, presentan mejores rendimientos para estimar precios sobre series temporales de salmón, esto en directa relación a las métricas obtenidas y que se encuentran presentes en el cuadro 6.1, esto en relación a los modelos tales como ARIMA, modelos clásicos de ML y modelos de DL con ajuste automático, dando respuesta a la hipótesis planteada en la Sección 1.

Cuadro 6.1: Modelos con resultados más altos de métricas

Modelo	Serie de Tiempo	MAE	MSE	R2
ARIMA	Serie Salar USA	2,130	2,600	0,580
K Neighbors Regressor	Serie Salar	0,849	1,445	0,391
Deep Learning (AutoKeras)	Serie Salar Congelado Europa	0,401	0,234	0,809
LSTM	Serie Salar	0,215	0,080	0,943
CNN 1D	Serie Salar	0,137	0,027	0,981

6.1. Trabajos futuros

La evaluación de los modelos consideró una visión limitada de los hiperparámetros disponibles y que pueden ser manipulados en cada una de las capas empleadas para la construcción de arquitecturas en cada modelo que se desprende de las diferentes series de tiempo empleadas, lo que se debe, a lo extenso de los experimentos y a una cantidad no menor de series utilizadas. Un trabajo futuro a considerar, es en ampliar los experimentos considerando otros hiperparámetros y valores en cada uno de los modelos implementados.

Otro aspecto a considerar, las métricas empleadas en registrar el desempeño de los modelos de DL en los distintos experimentos que se fueron ejecutados, permitiendo realizar comparaciones con diferentes hiperparámetros y sus valores. En este caso, las métricas no presentan información del aporte que realiza cada atributo en el desempeño de un modelo, este análisis proporcionaría un mejor entendimiento de los modelos y en este caso, si las variables utilizadas aportan en la implementación del modelo, ayudando a mejorar los esfuerzos en la obtención y tratamiento de los datos, a través de test de hipótesis a fin de detectar aquellos atributos que puedan aportar al modelo implementado.

Por otro lado, la combinación de modelos no fue considerado en esta tesis. Para el caso de los modelos CNN y LSTM se requiere una última capa *Dense*, que tiene como resultado solo un valor en la salida. La adición de una mayor cantidad de capas ocultas podría mejorar el desempeño de los modelos, tanto de los modelos LSTM como también los modelos CNN 1D. Se considera como trabajo a futuro, generar una combinación de modelos LSTM-CNN para generar estimaciones y cuantificar su desempeño estableciendo comparaciones entre modelos otros modelos que no son combinados.

Los experimentos se enfocaron en estimar el precio del salmón salar, en diferentes categorías y mercados de destino. Un trabajo a futuro consistiría en considerar otras variables asociadas al salmón como producto final, y que puedan robustecer un modelo de DL y poder generar comparaciones con otros mercados de destino y el comportamiento que tiene el precio del salmón según sus características.

El conjunto de datos original de 99.353 instancias considera información del salmón con ciertas características según códigos arancelarios. Como trabajo futuro es generar la detección automática estandarizando aquellos códigos arancelarios a fin de identificar productos derivados del salmón y que generan un mayor valor agregado al momento de generar las exportaciones.

Finalmente, el desarrollo de una plataforma basada en modelos predictivos de ML y DL podría generar un gran impacto en el sector, aportando en el análisis de datos, procesamiento, la correcta implementación y evaluación de modelos de diferentes características. Aportando a la toma de decisiones, facilitando un seguimiento y la evolución del mercado, estableciendo una nueva propuesta y que sea más eficiente frente a la toma de decisiones.

Anexo 1

Resumen métricas evaluadas en PyCaret

El cuadro 6.2 presenta el resumen de los experimentos ejecutados sobre la totalidad de las series temporales elaboradas, encontrar mejores resultados para MAE, MSE y R^2 .

Cuadro 6.2: Resumen Modelos clásicos con mejor rendimiento utilizando PyCaret

Serie de Tiempo	Modelo	MAE	MSE	R2
Serie Salar	K Neighbors Regressor	0,8491	1,4459	0,3912
	Random Forest Regressor	0,9001	1,5013	0,3885
	Gradient Boosting Regressor	0,9036	1,6479	0,3601
Serie Salar USA	Gradient Boosting Regressor	0,9352	1,6254	0,4365
	K Neighbors Regressor	0,9419	1,6743	0,3938
	Random Forest Regressor	0,9814	1,8157	0,3502
Serie Salar Congelado USA	Random Forest Regressor	0,5834	0,5317	0,4603
	AdaBoost Regressor	0,6002	0,5371	0,4504
	Gradient Boosting Regressor	0,6141	0,5821	0,4210
Serie Salar Fresco USA	Extra Trees Regressor	0,7349	0,7893	0,3943
	Gradient Boosting Regressor	0,6490	0,7359	0,3798
	K Neighbors Regressor	0,7728	0,8486	0,3640
Serie Salar Congelado Europa	AdaBoost Regressor	0,4102	0,2539	0,7338
	Extra Trees Regressor	0,4022	0,2928	0,7021
	Gradient Boosting Regressor	0,3907	0,2851	0,6842
Serie Salar Fresco Europa	Random Forest Regressor	0,6361	0,7226	0,0912
	Light Gradient Boosting Machine	0,6414	0,7592	0,0624
	AdaBoost Regressor	0,6641	0,7477	0,0463

Resumen métricas evaluadas en Serie Salar

El cuadro 6.3 presenta el resumen de los experimentos ejecutados sobre la Serie Salar y cada uno de los hiperparámetros utilizados y modificados, teniendo por objetivo, encontrar mejores resultados para MAE, MSE y R^2 .

Cuadro 6.3: Resumen hiperparámetros Serie Salmón Salar

Hiperparámetros	Evaluación	MAE	MSE	R2
Capas	2	0,215	0,080	0,943
	3	0,236	0,114	0,919
	4	0,313	0,153	0,892
	5	0,263	0,106	0,925
	6	0,236	0,093	0,934
	7	0,300	0,139	0,901
Neuronas	100	0,215	0,080	0,943
	300	0,314	0,161	0,886
	500	0,311	0,154	0,891
Función de activación	linear	0,215	0,080	0,943
	relu	0,277	0,180	0,872
	tanh	0,277	0,135	0,904
	sigmoid	0,252	0,119	0,916
	softmax	0,337	0,237	0,832
Dropout recurrente	0	0,215	0,080	0,943
	0,1	0,250	0,111	0,921
	0,2	0,257	0,124	0,912
	0,3	0,288	0,138	0,902
	0,4	0,245	0,119	0,916
	0,5	0,239	0,108	0,924
	0,6	0,266	0,126	0,911
	0,7	0,222	0,091	0,936
	0,8	0,316	0,176	0,876
0,9	0,310	0,156	0,889	
Dropout	0	0,215	0,080	0,943
	0,1	0,240	0,119	0,916
	0,2	0,239	0,108	0,924
	0,3	0,255	0,113	0,920
	0,4	0,248	0,101	0,928
Optimizador	Adam	0,215	0,080	0,943
	SGD	0,408	0,270	0,808
	RMSprop	0,379	0,211	0,851
	Adadelta	0,845	1,099	0,222
	Adamax	0,278	0,163	0,885

Función de activación

Los resultados que aquí se detallan provienen de la sección 5.5.2 y presenta los principales resultados obtenidos sobre cada una de las series temporales.

Cuadro 6.4: Resumen de métricas función de activación

Series de tiempo	Función de activación	MAE	MSE	R2
Serie Salar USA	linear	0,267	0,159	0,917
	relu	0,329	0,203	0,894
	tanh	0,273	0,160	0,916
	sigmoid	0,322	0,194	0,899
	softmax	0,280	0,154	0,920
Serie Salar Congelado USA	linear	0,363	0,278	0,751
	relu	0,545	0,544	0,514
	tanh	0,538	0,609	0,455
	sigmoid	0,693	0,765	0,316
	softmax	1,168	2,014	-0,802
Serie Salar Fresco USA	linear	0,511	0,644	0,644
	relu	1,349	0,061	0,061
	tanh	0,459	0,681	0,681
	sigmoid	0,246	0,829	0,829
	softmax	0,493	0,657	0,657
Serie Salar Congelado Europa	linear	0,375	0,329	0,493
	relu	0,306	0,189	0,709
	tanh	0,316	0,256	0,606
	sigmoid	0,241	0,105	0,838
	softmax	0,380	0,253	0,611
Serie Salar Fresco Europa	linear	1,213	3,039	-0,685
	relu	0,875	1,544	0,144
	tanh	1,022	2,782	-0,543
	sigmoid	0,832	1,500	0,168
	softmax	0,787	1,601	0,112

Resumen métricas Dropout

Cuadro 6.5: Resumen de métricas Dropout

Series de tiempo	Dropout	MAE	MSE	R2
Serie Salar USA	0	0,267	0,159	0,917
	0.1	0,356	0,215	0,888
	0.2	0,282	0,175	0,909
	0.3	0,281	0,167	0,913
	0.4	0,270	0,169	0,912
Serie Salar Congelado USA	0	0,363	0,278	0,751
	0.1	0,523	0,520	0,535
	0.2	0,478	0,360	0,678
	0.3	0,513	0,383	0,658
	0.4	0,529	0,440	0,606
Serie Salar Fresco USA	0	0,409	0,246	0,829
	0.1	0,491	0,363	0,747
	0.2	0,700	0,667	0,536
	0.3	0,521	0,474	0,670
	0.4	0,614	0,483	0,664
Serie Salar Congelado Europa	0	0,241	0,105	0,838
	0.1	0,228	0,095	0,854
	0.2	0,226	0,103	0,842
	0.3	0,209	0,081	0,875
	0.4	0,361	0,217	0,666
Serie Salar Fresco Europa	0	0,832	1,500	0,168
	0.1	0,933	1,873	-0,039
	0.2	0,894	1,725	0,044
	0.3	0,951	1,915	-0,062
	0.4	0,919	1,801	0,001

Resumen métricas Optimizador

Cuadro 6.6: Resumen de métricas Optimizador

Serie de tiempo	Optimizador	MAE	MSE	R2
Serie Salar USA	Adam	0,267	0,159	0,917
	SGD	0,473	0,347	0,819
	RMSprop	0,278	0,141	0,927
	Adadelta	1,014	1,689	0,118
	Adamax	0,335	0,203	0,894
Serie Salar Congelado USA	Adam	0,363	0,278	0,751
	SGD	0,616	0,612	0,452
	RMSprop	0,467	0,352	0,685
	Adadelta	1,069	1,758	-0,572
	Adamax	0,716	0,718	0,358
Serie Salar Fresco USA	Adam	0,409	0,246	0,829
	SGD	0,767	0,927	0,355
	RMSprop	0,661	0,593	0,587
	Adadelta	1,375	2,631	-0,832
	Adamax	0,639	0,570	0,603
Serie Salar Congelado Europa	Adam	0,209	0,081	0,875
	SGD	0,764	0,744	-0,145
	RMSprop	0,348	0,176	0,728
	Adadelta	0,784	0,874	-0,346
	Adamax	0,186	0,067	0,897
Serie Salar Fresco Europa	Adam	0,832	1,500	0,168
	SGD	1,096	2,143	-0,188
	RMSprop	0,789	1,246	0,309
	Adadelta	1,201	2,507	-0,390
	Adamax	0,940	1,517	0,159

Anexo 2

Códigos arancelarios asociado a salmón como un conjunto total en el dataset, sin distinción por especies.

Cuadro 6.7: Partida Arancelaria (302) Pescado fresco o refrigerado, excepto los filetes y demás carne de pescado de la partida 03.04

Partida Arancelaria	Glosa Arancelaria	Item SA
302	Pescado fresco o refrigerado, excepto los filetes	3021110
		3021120
		3021130
		3021190
		3021310
		3021320
		3021330
		3021340
		3021390
		3021410
		3021420
		3021430
		3021440
		3021490
3021900		

Cuadro 6.8: Partida Arancelaria (303) Pescado congelado, excepto los filetes y demás carne de pescado de la partida 03.04.

Partida Arancelaria	Glosa Arancelaria	Item SA
303	Pescado congelado, excepto los filetes	3031110
		3031120
		3031130
		3031140
		3031150
		3031190
		3031210
		3031220
		3031230
		3031240
		3031250
		3031290
		3031310
		3031320
		3031330
		3031340
		3031350
		3031390
		3031410
		3031420
		3031430
		3031440
3031490		
3031900		
3039110		
3039120		

Cuadro 6.9: Partida Arancelaria (304) Filetes y demás carne de pescado (incluso picada), frescos, refrigerados o congelados

Partida Arancelaria	Glosa Arancelaria	Item SA
304	Filetes y demás carne de pescado (incluso picada)	3044110
		3044120
		3044200
		3045200
		3048110
		3048120
		3048200
		3049941
		3049942
		3049943
		3049944
		3049951
		3049959

Cuadro 6.10: Partida Arancelaria (305) Pescado seco, salado o en salmuera; pescado ahumado, incluso cocido antes o durante el ahumado; harina, polvo y «pellets» de pescado, aptos para la alimentación humana

Partida Arancelaria	Glosa Arancelaria	Item SA
305	Pescado seco, salado o en salmuera; pescado ahumado	3053910
		3053920
		3054110
		3054120
		3054130
		3054140
		3054150
		3054160
		3054170
		3054180
		3054190
		3054310
		3054320
		3054330
		3054390
		3055920
		3055930
		3055940
		3055950
		3056911
		3056912
		3056913
		3056914
		3056915
		3056916
		3056919
		3056921
		3056922
		3056923
		3056924
3056925		
3056926		
3056929		

Anexo 3

Códigos arancelarios asociado a Salmón Salar como un conjunto total en el dataset.

Cuadro 6.11: Partida Arancelaria 302 Salmón Salar Pescado fresco o refrigerado, excepto los filetes y demás carne de pescado de la partida 03.04

Partida Arancelaria	Glosa Arancelaria	Item SA
302	Pescado fresco o refrigerado	3021410
		3021420
		3021430
		3021440
		3021490

Cuadro 6.12: Partida Arancelaria 303 Salmón Salar Pescado congelado, excepto los filetes y demás carne de pescado de la partida 03.04.

Partida Arancelaria	Glosa Arancelaria	Item SA
303	Pescado congelado	3031310
		3031320
		3031330
		3031340
		3031350
		3031350
		3039110

Cuadro 6.13: Partida Arancelaria 304 Salmón Salar Filetes y demás carne de pescado (incluso picada), frescos, refrigerados o congelados.

Partida Arancelaria	Glosa Arancelaria	Item SA
304	Filetes y demás carne de pescado	3044120
		3045200
		3048120
		3049929
		3049943
3049944		

Cuadro 6.14: Partida Arancelaria 305 Salmón Salar Pescado seco, salado o en salmuera; pescado ahumado, incluso cocido antes o durante el ahumado; harina, polvo y «pellets» de pescado, aptos para la alimentación humana.

Partida Arancelaria	Glosa Arancelaria	Item SA
305	Pescado seco, salado o en salmuera; pescado ahumado	3052010
		3053910
		3054140
		3054150
		3054160
		3054180
		3054190
		3055930
		3056911
		3056912
		3056913
		3056914
		3056915
		3056916
		3056919

Bibliografía

- [1] Instituto Nacional de Estadísticas, *Estudio de Caracterización de la Industria Salmonera*, 2019.
- [2] R. Santos, A. Pabon, W. Silva, H. Silva, and M. Pinho, *Population structure and movement patterns of blackbelly rosefish in the NE Atlantic Ocean (Azores archipelago)*, vol. 29, no. 3. 2020.
- [3] D. Bloznelis, “Short-term salmon price forecasting,” *J. Forecast.*, vol. 37, no. 2, pp. 151–169, 2018, doi: 10.1002/for.2482.
- [4] Z. Hu, Y. Zhao, and M. Khushi, “A survey of forex and stock price prediction using deep learning,” *Appl. Syst. Innov.*, vol. 4, no. 1, pp. 1–30, 2021, doi: 10.3390/ASI4010009.
- [5] W. Long, Z. Lu, and L. Cui, “Deep learning-based feature engineering for stock price movement prediction,” *Knowledge-Based Syst.*, vol. 164, pp. 163–173, 2019, doi: 10.1016/j.knosys.2018.10.034.
- [6] S. Ji, J. Kim, and H. Im, “A comparative study of bitcoin price prediction using deep learning,” *Mathematics*, vol. 7, no. 10, 2019, doi: 10.3390/math7100898.
- [7] M. Nikou, G. Mansourfar, and J. Bagherzadeh, “Stock price prediction using DEEP learning algorithm and its comparison with machine learning algorithms,” *Intell. Syst. Accounting, Financ. Manag.*, vol. 26, no. 4, pp. 164–174, 2019, doi: 10.1002/isaf.1459.
- [8] L. Pascual and E. Ruiz, *Predicción de series temporales basada en Machine Learning: aplicaciones económicas y financieras*, pp. 189–214.
- [9] B. Lim and S. Zohren, “Time-series forecasting with deep learning: A survey,” *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.*, vol. 379, no. 2194, 2021, doi: 10.1098/rsta.2020.0209.
- [10] S. Lim, S. Jun, Y. Park, and N. Kwon, “A deep learning-based time series model with missing value handling techniques to predict various types of liquid cargo traffic,” *Expert Syst. Appl.*, vol. 184, p. 115532, 2021, doi: 10.1016/j.eswa.2021.115532.
- [11] A. Burkov, *The Hundred-Page Machine Learning Book*, vol. 22, no. 2. 2020.
- [12] F. P. Cabrera, “Desarrollo de un modelo basado en Machine Learning para la predicción de la demanda de habitaciones y ocupación en el sector hotelero,” *Universidad Tecnológica de Bolívar*, 2014.
- [13] R. A. P. López, “Uso de Redes Neuronales Convolucionales Aplicado a Sentiment Analysis,” 2019.
- [14] H. Schulz, S. Shalev-shwartz, C. Science, S. Ben-david, and C. Science, *Understanding Machine Learning: from Theory to Algorithms*. 2011.
- [15] T. M. P. Mehryar Mohri, Afshin Rostamizadeh, Ameet Talwalkar - *Foundations of Machine Learning* (2012, Adaptive Computation and Machine Learning, vol. 0, no. 9783319056050. 2014.
- [16] C. Gao, E. Bompard, R. Napoli, and H. Cheng, “Price forecast in the competitive electricity market by support vector machine,” *Phys. A Stat. Mech. its Appl.*, vol. 382, no. 1, pp. 98–113, 2007, doi: 10.1016/j.physa.2007.03.050.
- [17] C.-J. L. Chih-Wei Hsu, Chih-Chung Chang, “Practical Guide to Support Vector Classification,” *Dep. Comput. Sci.*, vol. 17, no. 5, pp. 819–832, 2004.

- [18] K. Rathan, S. V. Sai, and T. S. Manikanta, “Crypto-currency price prediction using decision tree and regression techniques,” *Proc. Int. Conf. Trends Electron. Informatics, ICOEI 2019*, vol. 2019-April, no. Icoei, pp. 190–194, 2019, doi: 10.1109/icoei.2019.8862585.
- [19] C. C. Aggarwal, *Neural Networks and Deep Learning*. 2018.
- [20] N. J. Nilsson, “INTRODUCTION TO MACHINE LEARNING AN EARLY DRAFT OF A PROPOSED TEXTBOOK Department of Computer Science,” *Mach. Learn.*, vol. 56, no. 2, pp. 387–99, 2005, [Online]. Available: <http://www.ncbi.nlm.nih.gov/pubmed/21172442>.
- [21] Laura Flores Martínez, “Universidad Politécnica de Madrid Trabajo Fin de Máster Aplicación de Métodos de Deep Learning a un Sistema de Evaluación de Estudiantes,” Universidad Politécnica de Madrid, 2020.
- [22] A. Alonso Rodríguez, “Análisis de las series temporales a la luz de Deep Learning,” *Anu. jurídico y económico Ecur.*, no. 52, pp. 257–276, 2019.
- [23] F. Chollet, *Deep Learning with Python*. 2017.
- [24] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” *Nature*, vol. 323, no. 6088, pp. 533–536, 1986, doi: 10.1038/323533a0.
- [25] L. Bottou, “Large-Scale Machine Learning with Stochastic Gradient Descent,” *Proc. COMPS-TAT’2010*, pp. 3–4, 2010, doi: 10.1007/978-3-7908-2604-3.
- [26] L. Lausen, “GluonCV and GluonNLP: Deep Learning in Computer Vision and Natural Language Processing,” vol. 21, pp. 1–7, 2020.
- [27] E. Milli, “Electricity Price Prediction Using Encoder-Decoder Recurrent Neural Networks in Turkish Dayahead Market,” pp. 25–28.
- [28] T. L. Fine, “Feedforward Neural Network Methodology,” *IEEE Trans. Neural Networks*, vol. 12, no. 3, pp. 647–648, 2001, doi: 10.1109/TNN.2001.925573.
- [29] Z. Gao and X. Wang, “Deep learning,” *EEG Signal Process. Featur. Extr.*, pp. 325–333, 2019, doi: 10.1007/978-981-13-9113-2-16.
- [30] C. Fan, J. Wang, W. Gang, and S. Li, “Assessment of deep recurrent neural network-based strategies for short-term building energy predictions,” *Appl. Energy*, vol. 236, no. July 2018, pp. 700–710, 2019, doi: 10.1016/j.apenergy.2018.12.004.
- [31] S. Hochreiter, “Long Short-Term Memory,” vol. 1780, pp. 1735–1780, 1997.
- [32] F. F. Ting, Y. J. Tan, and K. S. Sim, “Convolutional neural network improvement for breast cancer classification,” *Expert Syst. Appl.*, vol. 120, pp. 103–115, 2019, doi: 10.1016/j.eswa.2018.11.008.
- [33] B. P. Andersen and P. E. de Lange, “Efficiency in the Atlantic salmon futures market,” *J. Futur. Mark.*, vol. 41, no. 6, pp. 949–984, 2021, doi: 10.1002/fut.22204.
- [34] A. Ntakaris, G. Mirone, J. Kannianen, M. Gabbouj, and A. Iosifidis, “Feature Engineering for Mid-Price Prediction with Deep Learning,” *IEEE Access*, vol. 7, pp. 82390–82412, 2019, doi: 10.1109/ACCESS.2019.2924353.
- [35] E. M. Rojas, M. R. Ramírez, B. Y. M. Lobato, H. B. R. Moreno, M. Del Consuelo Salgado Soto, and S. O. V. Nuñez, “Static model and neural networks in the prediction of price using time series,” *ACM Int. Conf. Proceeding Ser.*, pp. 2–7, 2018, doi: 10.1145/3302425.3302440.
- [36] S. Özen, V. Atalay, and A. Yazici, “Comparison of predictive models for forecasting timeseries data,” *ACM Int. Conf. Proceeding Ser.*, pp. 172–176, 2019, doi: 10.1145/3372454.3372482.
- [37] P. Lara-Benítez, M. Carranza-García, and J. C. Riquelme, “An Experimental Review on Deep Learning Architectures for Time Series Forecasting,” *Int. J. Neural Syst.*, vol. 31, no. 3, pp. 1–25, 2021, doi: 10.1142/S0129065721300011.

- [38] Gladilin Peter and Maria Matskevichus, "Hyperparameters Tuning for Machine Learning Models for Time Series Forecasting," 2019 Sixth Int. Conf. Soc. Networks Anal. Manag. Secur., no. 1973, pp. 2–43, 2019.
- [39] H. Liu and Z. Long, "An improved deep learning model for predicting stock market price time series," *Digit. Signal Process. A Rev. J.*, vol. 102, p. 102741, 2020, doi: 10.1016/j.dsp.2020.102741.
- [40] Andrés García-Florian, Cuauhtémoc López-Martín, Cornelio Yáñez-Márquez, and Alain Abran. Support vector regression for predicting software enhancement effort. *Information and Software Technology*, 97(January):99–109, 2018. ISSN 09505849. doi: 10.1016/j.infsof.2018.01.003. URL <https://doi.org/10.1016/j.infsof.2018.01.003>.
- [41] LeCun, Y., Bengio, Y. and Hinton, G. Deep learning. *Nature* 521, 436–444 (2015). <https://doi.org/10.1038/nature14539>
- [42] Jason Brownlee, *Long Short-Term Memory Networks With Python*, (2017).
- [43] Olmos Carlos 'Evaluación de Modelos Predictivos Basados en Deep Learning sobre Estimación de Recursos Minerales', 2021.
- [44] Kingma, Diederik P. and Ba, Jimmy, Adam: A Method for Stochastic Optimization, 2014, doi = 10.48550/ARXIV.1412.6980. URL: <https://doi.org/10.48550/arxiv.1412.6980>, arXiv.
- [45] S. Kiranyaz, T. Ince, R. Hamila and M. Gabbouj, "Convolutional Neural Networks for patient-specific ECG classification," 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), 2015, pp. 2608-2611, doi: 10.1109/EMBC.2015.7318926.
- [46] G'eron Aur'elien. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, 2019. ISBN 9781492032649.
- [47] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, y Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020. ISSN 0001-0782. doi:10.1145/3422622. URL <https://dl.acm.org/doi/10.1145/3422622>.
- [48] Z. Cui, W. Chen, Y. Chen, Multi-Scale Convolutional Neural Networks for Time Series Classification (2016), <https://arxiv.org/abs/1603.06995>.