



**EDUCACIÓN**  
SECRETARÍA DE EDUCACIÓN PÚBLICA



TECNOLÓGICO  
NACIONAL DE MÉXICO



INSTITUTO TECNOLÓGICO DE LA PAZ  
DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN  
MAESTRÍA EN SISTEMAS COMPUTACIONALES

# **MODELO PREDICTIVO DE SERIES TEMPORALES DE DATOS HIDROGRÁFICOS MEDIANTE APRENDIZAJE AUTOMÁTICO**

## **TESIS**

QUE PARA OBTENER EL GRADO DE  
MAESTRO EN SISTEMAS COMPUTACIONALES

PRESENTA:

MARCO ANTONIO PÉREZ NAVARRO

DIRECTORES DE TESIS:

DR. MARCO ANTONIO CASTRO LIERA  
DR. LEONARDO TENORIO FERNÁNDEZ

LA PAZ, BAJA CALIFORNIA SUR, MÉXICO, NOVIEMBRE 2022.



Instituto Tecnológico de La Paz  
División de Estudios de Posgrado e Investigación

La Paz, B.C.S., **12/ DICIEMBRE /2022**

DEPL\_MSC/070/2022

ASUNTO: Autorización de impresión

**C. MARCO ANTONIO PÉREZ NAVARRO,  
ESTUDIANTE DE LA MAESTRÍA EN  
SISTEMAS COMPUTACIONALES,  
P R E S E N T E .**

Con base en el dictamen de aprobación emitido por el Comité Tutorial de la Tesis denominada: **“MODELO PREDICTIVO DE SERIES TEMPORALES DE DATOS HIDROGRÁFICOS MEDIANTE APRENDIZAJE AUTOMÁTICO”**, mediante la opción de tesis (Proyectos de Investigación), entregado por usted para su análisis, le informamos que se **AUTORIZA** la impresión.

**ATENTAMENTE**

*Excellencia en Educación Tecnológica*

  
**JUDITH GUADALUPE MARTÍNEZ TIRADO,  
JEFA DE LA DIV. DE ESTUDIOS DE POSGRADO E INV.**



c.c.p. Depto. de Servicios Escolares  
c.c.p. Archivo.

JGMT/icl\*



## DICTAMEN DEL COMITÉ TUTORIAL

La Paz, B.C.S., **06/DICIEMBRE/ 2022**

**JUDITH GUADALUPE MARTÍNEZ TIRADO,  
JEFA DE LA DIVISIÓN DE ESTUDIOS DE  
POSGRADO E INVESTIGACIÓN,  
P R E S E N T E.**

Por medio del presente, enviamos a usted dictamen del Comité Tutorial de tesis para la obtención del grado de Maestro, con los siguientes datos generales:

No. de Control M21310001	Nombre MARCO ANTONIO PÉREZ NAVARRO
Maestría en:	SISTEMAS COMPUTACIONALES
Título de la tesis: MODELO PREDICTIVO DE SERIES TEMPORALES DE DATOS HIDROGRÁFICOS MEDIANTE APRENDIZAJE AUTOMÁTICO	
<b>DICTAMEN:</b> Se autoriza el trabajo de investigación, en virtud de que realizó las correcciones correspondientes conforme a las observaciones planteadas por este Comité Tutorial.	

**Atentamente.  
El Comité Tutorial**

  
MC. JORGE ENRIQUE LUNA TAYLOR

  
MSC. ILIANA CASTRO LIERA

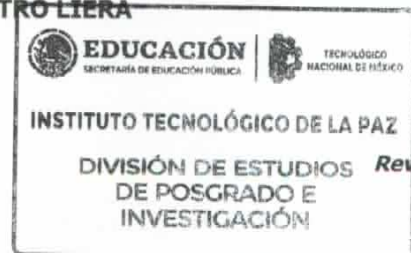
  
DR. LEONARDO TENORIO FERNÁNDEZ

  
MSC. EMMANUEL ROMERO QUINTERO

  
DR. MARCO ANTONIO CASTRO LIERA

c.c.p. Coordinador de la Maestría.  
c.c.p. Departamento de Servicios Escolares.  
c.c.p. Estudiante.

**ITLP-DEPI-RTT-08**



Boulevard Forjadores de B.C.S. #4720, Col. 8 de Octubre 1ra Sección, C.P. 23080, La Paz,  
B.C.S., Tel. (612) 12 10424 e-mail: [depi\\_paz@tecnm.mx](mailto:depi_paz@tecnm.mx) [tecnm.mx](http://tecnm.mx) | [lapaz.tecnm.mx](http://lapaz.tecnm.mx)



# Dedicatoria

*A mi compañera de vida Verónica,  
a mis hijos Eduardo Antonio,  
Marco Alexis y Alejandro Leví.  
¡Son mi motor e inspiración!*

# Agradecimientos

A mi esposa e hijos por su incondicional apoyo, comprensión y paciencia durante el tiempo dedicado a la Maestría. A mi madre y hermanos, porque se que siempre puedo contar con ustedes.

A los integrantes de mi comité tutorial MSC Iliana Castro Liera, Dr. Marco Antonio Castro Liera, Dr. Leonardo Tenorio Fernández, MC Jorge Enrique Luna Taylo y MSC Emmanuel Romero Quintero, por sus enseñanzas, disposición, orientación e invaluable apoyo durante todo este proceso. Al Dr. Saúl Martínez Díaz y al Dr. Israel Marco Santillán Méndez por haber sido parte de mi formación como Maestro en Sistemas Computacionales.

Al Consejo Nacional de Ciencia y Tecnología (CONACYT) por el otorgamiento de la beca, la cual me permitió dedicarme de tiempo completo a los estudios de maestría.

Al Instituto Tecnológico de La Paz, mi alma máter, por ofrecer opciones para la superación continua de la comunidad tecnológica y la comunidad en general.

# Resumen

En este trabajo se implementa un modelo basado en aprendizaje automático, utilizando redes neuronales artificiales, para completar series temporales de datos hidrográficos en un polígono dado. El área de estudio seleccionada está delimitada por un polígono irregular, abarcando parte de la región central y sur del Golfo de México. Se utilizaron datos *in situ* descargados de la base de datos oceánica mundial (WOD18), seleccionando los datos correspondientes al periodo comprendido entre los años 2013 al 2021, y como primera etapa, datos con profundidad de cero a diez metros. El modelo fue implementado en Keras con Tensorflow, en el lenguaje Python. En la metodología propuesta, se utilizó la librería Talos para determinar la mejor combinación de número de capas ocultas y número de neuronas por capa oculta para el modelo. Se realizó la predicción con una escala temporal mensual y con una escala espacial de una cuadrícula de  $2^\circ \times 2^\circ$ . La división del conjunto de datos se efectuó de forma aleatoria, repitiendo el procedimiento de división de los datos y entrenamiento del modelo diez veces y posteriormente seleccionando el modelo con el menor valor en la función de pérdida con los datos de prueba. El modelo fue evaluado mediante el error cuadrático medio (MSE) y el coeficiente de determinación ( $R^2$ ), obteniendo para el MSE un valor global de 0.0095, mientras que para  $R^2$  fue de 0.9985, los cuales superan a resultados de trabajos similares donde reportan el uso de redes neuronales artificiales.

# Abstract

In this work, a model based on machine learning is implemented, using artificial neural networks, to complete time series of hydrographic data in a given polygon. The study area is delimited by an irregular polygon, encompassing part of the central and southern region of the western Gulf of Mexico. Data downloaded from the World Ocean Database (WOD18) were used, selecting the data corresponding to the period between 2013 and 2021, and, data with depth from zero to ten meters. The model was implemented using Keras on Tensorflow in the Python programming. In the proposed methodology, the Talos library was used to determine the best combination of number of hidden layers and number of neurons per hidden layer. The prediction was made with a monthly time scale and a spacial scale of a  $2^\circ \times 2^\circ$  grid. The dataset was divided randomly, repeating the procedure of data division and training of the model ten times and later selecting the model with the lowest value in the loss function with the test dataset. The model was evaluated using the mean squared error (MSE) and the coefficient of determination ( $R^2$ ), getting a global value of MSE of 0.0095, and for  $R^2$  getting a value of 0.9985, both results exceed the reported results of similar works where artificial neural networks were used.

# Índice general

<b>1. Introducción</b>	<b>1</b>
1. 1. Antecedentes . . . . .	1
1. 2. Descripción del problema . . . . .	4
1. 3. Objetivos . . . . .	5
1. 3.1. Objetivo general . . . . .	5
1. 3.2. Objetivos específicos . . . . .	5
1. 4. Justificación . . . . .	5
1. 5. Alcance y limitaciones . . . . .	6
1. 6. Hipótesis . . . . .	6
<b>2. Marco teórico</b>	<b>7</b>
2. 1. Geometría computacional . . . . .	7
2. 1.1. Cálculo de perímetros . . . . .	7
2. 1.2. Cálculo de áreas . . . . .	8
2. 1.3. Algoritmo punto en polígono . . . . .	8
2. 2. Análisis exploratorio de datos . . . . .	9
2. 3. Bases de datos . . . . .	9
2. 3.1. Datos Argo . . . . .	9
2. 3.2. Datos de la base de datos oceánica mundial (WOD) . . . . .	10
2. 4. Series de tiempo . . . . .	10
2. 4.1. Modelos estadísticos para series de tiempo . . . . .	12
2. 4.2. Aprendizaje automático y aprendizaje profundo para series temporales . . . . .	12
2. 5. Aprendizaje automático ( <i>Machine Learning</i> ) . . . . .	13
2. 5.1. Introducción a las Redes Neuronales Artificiales . . . . .	14



2. 5.2. El Perceptrón . . . . .	14
2. 5.3. Perceptrón multicapa (MLP) . . . . .	17
2. 5.4. Función de activación . . . . .	20
2. 5.5. Algoritmos optimizadores . . . . .	21
2. 5.6. Redes Neuronales Recurrentes (RNN) . . . . .	21
2. 5.7. Redes Neuronales Convolucionales (CNN) . . . . .	23
2. 5.8. Arquitecturas combinadas . . . . .	27
2. 6. Antecedentes del pronóstico de series temporales mediante aprendizaje automático	29
<b>3. Metodología</b>	<b>33</b>
3. 1. Selección de datos . . . . .	33
3. 1.1. Datos Argo . . . . .	33
3. 1.2. Datos de la base de datos oceánica mundial (WOD) . . . . .	33
3. 2. Análisis exploratorio de datos . . . . .	35
3. 3. Área de estudio . . . . .	35
3. 4. Modelos para predicción espacial . . . . .	35
3. 4.1. Selección del número de capas ocultas y neuronas por capa oculta . . . . .	36
3. 4.2. Implementación de los modelos de RNA . . . . .	37
3. 4.3. Selección de los datos de entrenamiento, validación, y prueba . . . . .	38
3. 4.4. Entrenamiento de los modelos de RNA . . . . .	39
3. 4.5. Evaluación de los modelos de RNA . . . . .	39
3. 4.6. Predicción espacial en nuevas ubicaciones . . . . .	40
<b>4. Resultados</b>	<b>41</b>
4. 1. Conjuntos de datos . . . . .	41
4. 2. Modelo de Red Neuronal Artificial . . . . .	47
4. 3. Resultados del entrenamiento . . . . .	47
4. 4. Evaluación de resultados . . . . .	51
<b>5. Conclusiones</b>	<b>52</b>
<b>A. Mediciones mensuales</b>	<b>53</b>

B. Predicciones espaciales mensuales	58
Bibliografía	58

# Índice de figuras

1.1. Distribución de flotadores operacionales Argo . . . . .	4
2.1. Representación del algoritmo basado en la fórmula de la lazada . . . . .	8
2.2. Representación del algoritmo punto en polígono . . . . .	9
2.3. Unidad lógica de umbral . . . . .	15
2.4. Ejemplo de Perceptrón . . . . .	16
2.5. Ejemplo de MLP . . . . .	17
2.6. MLP con Mecanismo de Atención . . . . .	20
2.7. Arquitectura simple de RNN . . . . .	22
2.8. “ <i>Unrolling</i> ” de una RNN . . . . .	22
2.9. Arquitectura típica de una CNN . . . . .	24
2.10. Reduciendo la dimensionalidad usando un <i>stride</i> de 2 . . . . .	25
2.11. Proceso de convolución . . . . .	26
2.12. Capas de convolución con múltiples mapas de características . . . . .	26
2.13. Agrupación, aplicando un filtro de $2 \times 2$ , y un <i>stride</i> de 2 . . . . .	27
2.14. Arquitectura del modelo LSTNet modificado . . . . .	28
3.1. Histograma de las mediciones por año . . . . .	34
3.2. Áreas propuestas . . . . .	36
4.1. Gráficas de dispersión de temperatura . . . . .	41
4.2. Gráficas de dispersión de presión, temperatura y salinidad, respecto al tiempo . . . . .	42
4.3. Gráficas de cajas de temperatura mensual . . . . .	43
4.4. Gráficas de dispersión de temperatura mensual . . . . .	43
4.5. Gráficas de dispersión de temperatura y salinidad práctica . . . . .	44

4.6. Distribución de mediciones de cada área propuesta . . . . .	46
4.7. Área de estudio seleccionada . . . . .	46
4.8. Red Neuronal Pre-alimentada . . . . .	47
4.9. Gráfica de dispersión de temperaturas observadas vs predichas . . . . .	49
4.10. Gráfica de dispersión relacional . . . . .	49
4.11. Diagramas de cajas y bigotes del error absoluto mensual . . . . .	50
A.1. Mediciones del año 2013 con profundidad de 0 a 10 metros. “count” indica el número de mediciones disponibles, “T min” el valor mínimo, “T max” el valor máximo . . . . .	53
A.2. Mediciones de los años 2014 y 2015 con profundidad de 0 a 10 metros. “count” indica el número de mediciones disponibles, “T min” el valor mínimo, “T max” el valor máximo . . . . .	54
A.3. Mediciones de los años 2016 y 2017 con profundidad de 0 a 10 metros. “count” indica el número de mediciones disponibles, “T min” el valor mínimo, “T max” el valor máximo . . . . .	55
A.4. Mediciones de los años 2018 y 2019 con profundidad de 0 a 10 metros. “count” indica el número de mediciones disponibles, “T min” el valor mínimo, “T max” el valor máximo . . . . .	56
A.5. Mediciones de los años 2020 y 2021 con profundidad de 0 a 10 metros. “count” indica el número de mediciones disponibles, “T min” el valor mínimo, “T max” el valor máximo . . . . .	57
B.1. Predicciones del año 2013. “count” indica el número de mediciones disponibles, “T min” el valor mínimo, “T max” el valor máximo . . . . .	58
B.2. Predicciones de los años 2014 y 2015. “count” indica el número de mediciones disponibles, “T min” el valor mínimo, “T max” el valor máximo . . . . .	59
B.3. Predicciones de los años 2016 y 2017. “count” indica el número de mediciones disponibles, “T min” el valor mínimo, “T max” el valor máximo . . . . .	60
B.4. Predicciones de los años 2018 y 2019. “count” indica el número de mediciones disponibles, “T min” el valor mínimo, “T max” el valor máximo . . . . .	61

B.5. Predicciones de los años 2020 y 2021. “count” indica el número de mediciones disponibles, “T min” el valor mínimo, “T max” el valor máximo . . . . . 62

# Índice de tablas

2.1. Conjuntos de datos (Dataset) de la WOD . . . . .	11
3.1. Áreas propuestas . . . . .	35
4.1. Perímetro calculado para cada área . . . . .	45
4.2. Área calculada para cada área propuesta . . . . .	45
4.3. Densidad de mediciones de cada área propuesta . . . . .	45
4.4. Hiperparámetros de los modelos de RNA para predicción espacial . . . . .	48
4.5. Resultados de las métricas . . . . .	51

# Capítulo 1

## Introducción

### 1. 1. Antecedentes

El océano cubre el 71 % de la superficie del planeta y está ligado al sustento de la vida de la humanidad de diferentes formas, las cuales van desde modular el clima hasta proveer una variedad de beneficios socio-económicos, culturales y ambientales.

De acuerdo al documento [1], el crecimiento de la población y el incremento en la concentración de dióxido de carbono en la atmósfera han aumentado exponencialmente con la industrialización. Lo que ha provocado un clima cambiante y la elevación de la temperatura del océano. Cada año, aproximadamente una cuarta parte del exceso de emisiones de dióxido de carbono son absorbidas por el océano, ayudando a disminuir la tasa de calentamiento atmosférico, pero con el desafortunado efecto secundario de hacer que el océano se vuelva más ácido, lo que afecta a una gran cantidad de organismos. El reciente aumento en la temperatura del océano ha dado lugar a un aumento constante del nivel del mar con la consecuente expansión del volumen de agua de mar y la adición de agua a las cuencas oceánicas debido al derretimiento del hielo. Además, el calentamiento del océano reduce la solubilidad del oxígeno, fundamental para la mayor parte de la vida oceánica.

El océano también se considera un importante impulsor de la economía mundial, ya que a través de él se transporta más del 90 % del comercio mundial y da sustento al 40 % de la humanidad que vive a menos de 100 Km de la costa. Actualmente, el creciente impacto del cambio climático está haciendo que las observaciones, investigaciones y servicios del océano sean más críticos que nunca [2].

El programa de la UNESCO [3] cuenta con un grupo de coordinación de observaciones (*Observation Coordination Group - OCG*) que se encarga de revisar, asesorar y coordinar las redes mundiales de observación. Uno de los propósitos del OCG es el crear una comunidad de investigación internacional que desarrolle conocimientos y soluciones para la próxima década. Un elemento clave para este objetivo es promover el intercambio de datos y metadatos internacionales y capacidades de monitoreo a través de *OceanOPS*. La *OceanOPS* es un sistema de observación oceánica global que ocupa un lugar único como punto focal para la coordinación de información y metadatos que fluyen de redes globales de observaciones *in situ*.

Otra fuente de información oceánica importante es la base de datos mundial de océanos (WOD, por sus siglas en inglés) producida por el Centro Nacional para Información Ambiental (NCEI) de la Administración Nacional Oceánica y Atmosférica (NOAA) de los Estados Unidos. La WOD es la colección más grande de perfiles de datos oceánicos con formato uniforme, control de calidad y disponibles públicamente. Es el resultado de un esfuerzo coordinado de más de veinte años para incorporar datos de instituciones, agencias, investigadores independientes, e iniciativas de recuperación de datos en una sola base de datos [4].

La versión más reciente de la base de datos WOD es la WOD18, la cual es una base de datos de calidad controlada científicamente de mediciones oceanográficas históricas *in situ* de la superficie y sub-superficie. La WOD18 incluye mediciones *in situ* de temperatura, salinidad, oxígeno disuelto, nutrientes inorgánicos disueltos, clorofila, por citar algunos [5].

Una de las fuentes de información de la base de datos WOD, es la base de datos conformada por las mediciones de los flotadores Argo. Argo es un programa internacional que recopila información del océano usando una flota de instrumentos robóticos (flotadores o Perfiladores Hidrográficos Autónomos (PHA)) que se mueven a la deriva con las corrientes oceánicas y entre la superficie y un nivel medio del agua [6]. Los datos que Argo recopila describen la temperatura y salinidad del agua, y algunos de los flotadores miden otras propiedades bioquímicas del océano. La principal razón para recopilar esos datos es ayudar a comprender el papel del océano en el clima de la tierra y así poder realizar estimaciones mejoradas de cómo cambiará en el futuro. Al año 2020 Argo recopilaba 12,000 perfiles de datos cada mes (400 por día).

A medida que se han realizado avances tecnológicos Argo ha ampliado su diseño original para incluir mares marginales, zonas de hielo-marino, y mediciones hasta el fondo del océano. Además, hay nuevos sensores que miden parámetros biogeoquímicos y a mayor profundidad. El



nuevo diseño, Argo2020, es global y multidisciplinario. Cuando el proyecto Argo sea completado, contará con 4,000 flotadores de los cuales 1,200 serán de Presión Temperatura y Salinidad (CTD), denominados flotadores núcleo, y 1,000 serán biogeoquímicos (BGC), los cuales registran valores de las siguientes variables: oxígeno, nitrato, pH, clorofila, partículas suspendidas e irradiación descendente . Todos los flotadores combinados proporcionarán más de 100,000 perfiles de temperatura y salinidad (T/S) y mediciones de velocidad por año, distribuidos en los océanos con un espaciamiento promedio de 3 grados [6].

Las misiones que comprenden el programa Argo son las siguientes:

- *Core Argo*: flotadores que realizan mediciones de temperatura, salinidad y presión, hasta una profundidad de 2,000 metros.
- *Deep Argo Mission*: flotadores que realizan mediciones de temperatura, salinidad y presión, hasta una profundidad de 6,000 metros.
- *BGC-Argo Mission*: flotadores que realizan mediciones de temperatura, salinidad, presión y oxígeno, pH, nitrato, clorofila, entre otros parámetros, hasta una profundidad de 2,000 metros.

Cada flotador Argo es lanzado desde un barco. El peso del flotador es cuidadosamente ajustado de forma tal que, a medida que se hunde, eventualmente se estabiliza a un nivel preestablecido, usualmente a 1,000 metros. Diez días después, una bomba interna impulsada por batería transfiere aceite entre un depósito dentro del flotador y un depósito exterior. Esto hace que el flotador primero descienda hasta 2,000 metros y luego regrese a la superficie midiendo las características hidrográficas del océano a medida que asciende. Ya en la superficie, el flotador transmite los datos y su posición a los satélites y de ahí se transmite a las estaciones receptoras en tierra. Posteriormente el flotador repite el ciclo de 10 días hasta que se agote su batería.

Las mediciones de los flotadores son enviados a centros regionales de datos donde se someten a rigurosos controles de calidad y luego se pasan a dos centros de datos globales desde donde cualquiera que desee utilizarlos puede acceder a ellos. En la Figura 1.1 se puede observar la distribución de flotadores *Argo*, así como la cantidad de flotadores operacionales en octubre de 2022.

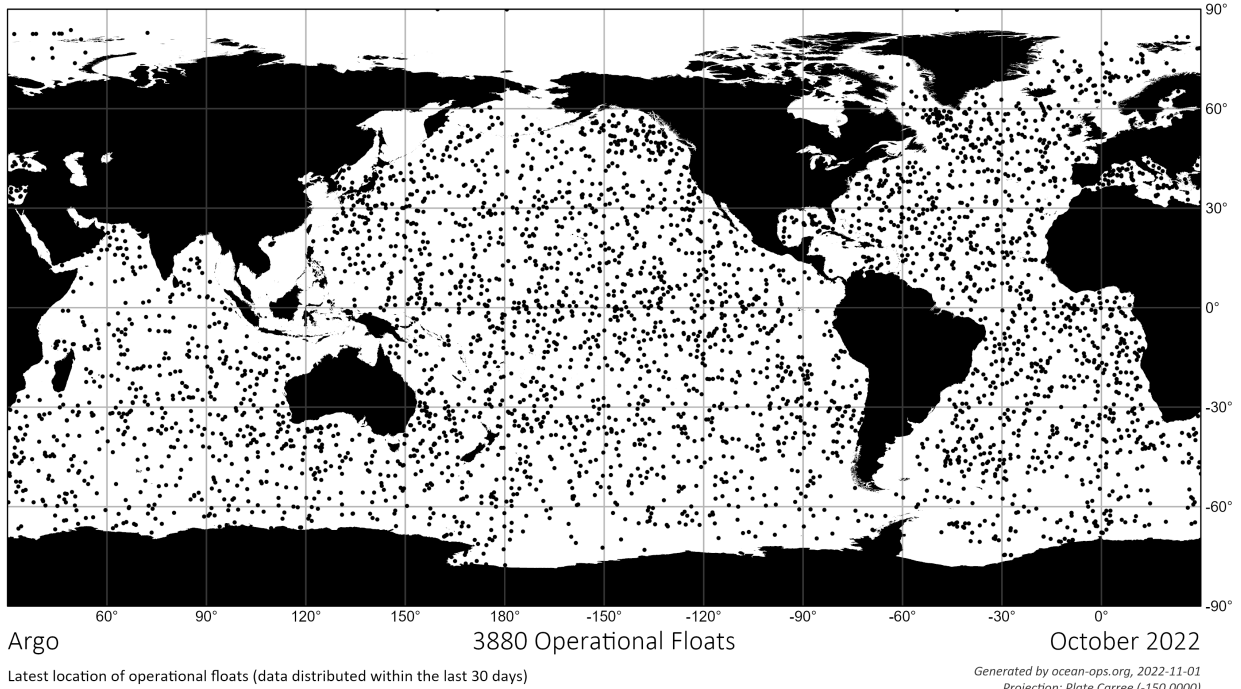


Figura 1.1: Distribución de flotadores operacionales Argo

## 1. 2. Descripción del problema

Hoy en día, la tecnología de observación de los océanos continua progresando, haciendo posible contar con mayor volumen de información de parámetros hidrográficos de diferentes fuentes, como información recopilada por satélites e información tomada *in situ*. De esta última fuente de información, el proyecto internacional Argo, desempeña un rol muy importante, ya que desde su inicio en el año 2000 y hasta la fecha, ha desplegado perfiladores hidrográficos autónomos en los océanos del mundo, teniendo, al momento, cerca de 3,900 dispositivos operativos, entre los que se encuentran perfiladores núcleo (CTD) y biogeoquímicos; sin embargo, existe evidencia documental, como los artículos de Portela et al. [7] y de Romero et al. [8], de la relativa falta de mediciones *in situ* en algunas zonas del océano, como por ejemplo la parte central y sur del Golfo de México, y en algunas otras áreas de la zona económica exclusiva (ZEE) de México.

Esta relativa falta de mediciones *in situ* da como resultado series temporales incompletas, es decir, con datos faltantes. El no poder contar con series temporales completas, vuelve más complejo el desarrollo y la implementación de modelos predictivos basados en aprendizaje automático.

Por lo anterior, en este proyecto de tesis se plantea el desarrollo de una metodología mediante la cual sea posible el poder completar la información faltante en las series temporales de datos hidrográficos por medio de la implementación de modelos de aprendizaje automático, así como con la búsqueda de bases de datos adicionales a la del proyecto Argo. Posteriormente, utilizar las series temporales para alimentar modelos predictivos de aprendizaje automático.

## **1. 3. Objetivos**

### **1. 3.1. Objetivo general**

Desarrollar un modelo de aprendizaje automático, implementando algoritmos de redes neuronales artificiales, mediante el cual se estimen los datos faltantes de series temporales de mediciones *in situ* de un polígono dado, y sea posible realizar predicciones de parámetros hidrográficos a partir de las series temporales.

### **1. 3.2. Objetivos específicos**

- Obtener, analizar y pre-procesar los conjuntos de datos de mediciones *in situ*.
- Determinar el modelo de aprendizaje automático con el mejor desempeño con base en los valores de las métricas que sean seleccionadas.
- Implementar el algoritmo, realizar pruebas, validar los resultados obtenidos.

## **1. 4. Justificación**

El poder contribuir con una metodología que permita generar información altamente confiable de las condiciones futuras, a corto, mediano, y largo plazo, motivan el desarrollo del presente trabajo. Con la certeza de que servirá como una herramienta para la comunidad científica dedicada al estudio de las condiciones de nuestros océanos y que su uso derive en información útil para las personas y organismos responsables de la toma de decisiones que permitan frenar y revertir los efectos adversos que el cambio climático provoca en los océanos y por consiguiente en nuestro planeta.

El uso de las redes neuronales artificiales (técnicas del aprendizaje automático y del aprendizaje profundo) han demostrado producir resultados que mejoran a los de otros métodos, en la estimación de los valores faltantes en series temporales. Además, las redes neuronales artificiales, han sido ampliamente utilizadas para hacer predicciones de condiciones futuras en diferentes campos de estudio.

## 1. 5. Alcance y limitaciones

- Las series temporales serán del parámetro de temperatura *in situ*.
- El modelo de aprendizaje automático se implementará utilizando librerías existentes y de uso libre basadas en lenguaje Python, como Keras con TensorFlow, scikit-learn, Talos, entre otras.
- Las bases de datos serán las disponibles de forma gratuita, siendo las principales la del proyecto Argo, así como la base de datos mundial (WOD18).

## 1. 6. Hipótesis

La información faltante en las series temporales de datos hidrográficos puede ser estimada, con un alto grado de confiabilidad, implementado modelos de aprendizaje automático de redes neuronales artificiales (RNA), y posteriormente hacer predicciones de condiciones futuras, mejorando los resultados de trabajos de investigación realizados a la fecha que utilicen datos hidrográficos y RNA.

# Capítulo 2

## Marco teórico

### 2. 1. Geometría computacional

#### 2. 1.1. Cálculo de perímetros

La fórmula del semiverseno (*haversine formula*) proporciona la distancia del círculo mayor entre dos puntos sobre la superficie del globo conociendo su longitud y latitud [9]. Esta fórmula se basa en la restricción de que considera que la tierra es esférica eliminando el factor que la tierra es ligeramente elíptica. Por lo tanto, el cálculo de la distancia de un punto a otro de la superficie terrestre se ve afectado por un cierto grado de curvatura [10].

La fórmula del semiverseno sirve para encontrar la distancia entre dos puntos mediante los cálculos 2.1 y 2.2:

$$a = \sin^2\left(\frac{lat2 - lat1}{2}\right) + \cos(lat1) * \cos(lat2) * \sin^2\left(\frac{lon2 - lon1}{2}\right) \quad (2.1)$$

$$d = \sqrt{a}R \quad (2.2)$$

Donde:

- lat1 y lon1 representan la latitud y longitud del punto 1,
- lat2 y lon2 representan la latitud y longitud del punto 2,
- R representa el radio de la tierra de 6371 Km,
- d es la distancia en kilómetros.

## 2. 1.2. Cálculo de áreas

En el presente trabajo se utilizó la fórmula del área de Gauss, también conocida como fórmula de la lazada (*shoelace formula*), para el cálculo de área de polígonos. Fue propuesta por Gauss en 1795, y es llamada de la lazada debido a que se asemeja al cálculo del producto cruzado de la matriz. La idea del algoritmo que implementa esta fórmula es dividir el polígono en triángulos y calcular la suma del área de todos los triángulos formados [11]. La ecuación 2.3 representa la fórmula para el cálculo de áreas. Las Figuras 2.1(a) y 2.1(b) ejemplifican la idea detrás del algoritmo basado en la fórmula de la lazada. La imagen de la izquierda muestra como es dividido el polígono. La imagen de la derecha ilustra el cálculo del producto cruzado.

$$A = \frac{1}{2} \sum_{i=1}^n [x_i * (y_{i+1} - y_{i-1})] \quad (2.3)$$

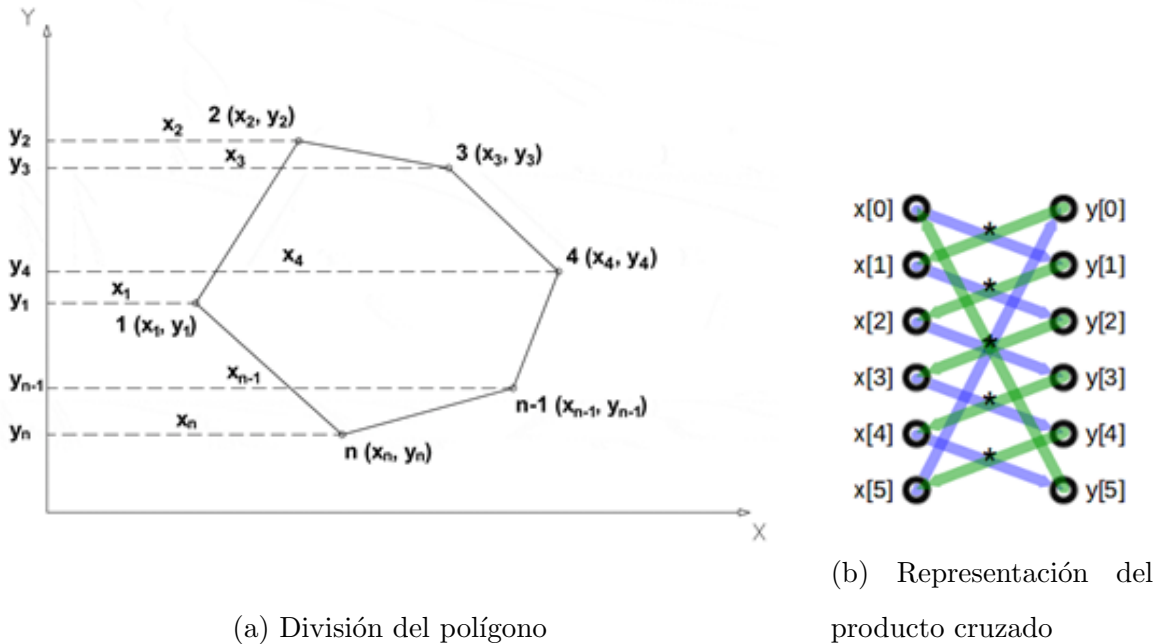


Figura 2.1: Representación del algoritmo basado en la fórmula de la lazada

## 2. 1.3. Algoritmo punto en polígono

El algoritmo punto en polígono es también conocido como algoritmo par-impar, o número de cruces, o de Jordan. La versión del algoritmo utilizado en este trabajo está basado en el teorema de la curva de Jordan, formulado y probado por Camille Jordan en 1882 [12].

El método utilizado consiste en trazar una línea horizontal desde un punto de prueba, y contar cuantas aristas o lados del polígono cruza. En la Figura 2.2 se representa el algoritmo, donde, los puntos color verde y rojo son puntos de prueba. La condición inicial es considerarlos fuera, y en cada cruce de una arista la condición cambia entre dentro y fuera. De esta forma se determina que el punto verde esta fuera y el punto rojo esta dentro del polígono.

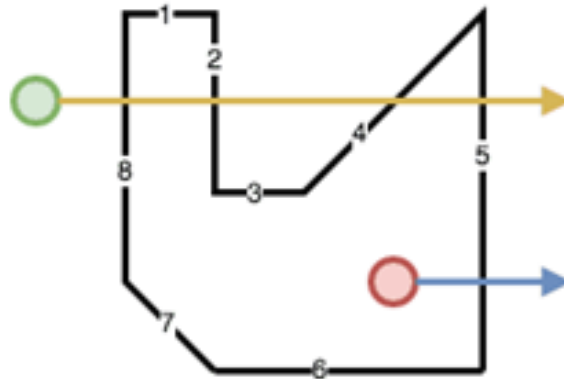


Figura 2.2: Representación del algoritmo punto en polígono

## 2. 2. Análisis exploratorio de datos

El análisis exploratorio de datos (EDA) ayuda a analizar los conjuntos de datos para resumir sus características estadísticas haciendo énfasis en aspectos clave, como, medidas de tendencia central (media, moda y mediana), medidas de dispersión (desviación estándar y varianza), la forma de la distribución y la existencia de valores atípicos. Con este análisis podemos conocer acerca del contenido y características del conjunto de datos, así como su tamaño. Es posible encontrar valores faltantes, o encontrar también, posibles relaciones entre los datos [13].

## 2. 3. Bases de datos

### 2. 3.1. Datos Argo

La fuente principal de datos para el presente trabajo de investigación es el proyecto Argo. Actualmente con cerca de 3,900 flotadores activos, proporciona una cantidad considerable de información hidrográfica.

El conjunto de datos completo de Argo está disponible en tiempo real y en modo diferido en los Centros Globales de Datos (GDACs), los cuales posibilitan el acceso a los datos mediante direcciones FTP, dirección de internet de los GDACs, así como también desde el sitio Argo DOI (Digital Object Identifier) en donde se actualizan los datos de forma mensual.

Los datos Argo pueden ser obtenidos en el sitio [14], donde se tienen disponibles copias de solo lectura de las bases de datos (*snapshots*), en formato NetCDF, publicadas mensualmente.

### **2. 3.2. Datos de la base de datos oceánica mundial (WOD)**

La WOD es una base de datos de calidad controlada de mediciones oceanográficas de la superficie y sub-superficie. Incluye mediciones *in situ* de temperatura, salinidad, oxígeno disuelto, nutrientes inorgánicos disueltos, clorofila, y algunos parámetros más. Producida por el Centro Nacional de Información Ambiental (NCEI) de la Administración Nacional Oceánica y Atmosférica (NOAA) de los Estados Unidos. La versión más reciente es la base de datos oceánica mundial 2018 (WOD18), contiene 3,560 millones de mediciones de perfiles individuales, de los cuales 1,950 millones son de temperatura y 1,130 millones de salinidad.

Los datos en la WOD son almacenados en once conjuntos de datos, cada uno representando un grupo similar de probetas oceanográficas como se muestra en la Tabla 2.1.

## **2. 4. Series de tiempo**

Las series de tiempo también son conocidas como series temporales o series cronológicas. Se definen como una secuencia orientada al tiempo o cronológica de observaciones sobre una variable de interés [15]. Una característica intrínseca de una serie temporal es que, típicamente, observaciones adyacentes son dependientes [16].

El análisis de series temporales es el esfuerzo de extraer información resumida y estadística significativa de puntos organizados en orden cronológico, y se realiza para diagnosticar comportamientos pasados y también para predecir comportamientos futuros [17]. El análisis de series temporales se ocupa de las técnicas para el análisis de la dependencia de observaciones adyacentes [16].



Tabla 2.1: Conjuntos de datos (Dataset) de la WOD

[Dataset] - Fuente
[OSD (Ocean Station Data)] - Mediciones realizadas desde barcos de investigación estacionarios utilizado termómetros de inversión y muestras de agua tomadas de botellas lanzadas a profundidades de interés en la columna de agua.
[CTD (Conductivity-Temperature-Depth)] - Instrumentos que combinan un sensor de presión, un dispositivo de medición de temperatura, y un sensor de conductividad.
[MBT (Mechanical Bathythermograph)] - Instrumento que proporciona una estimación de la temperatura en función de la profundidad.
[XBT (Expendable Bathythermograph)] - Dispositivos con forma de torpedo unido a un carrete de alambre de cobre. Lanzado desde un barco en movimiento. Estima la temperatura midiendo la resistencia en un semiconductor. Para grabar la información se envía a través del cable de cobre.
[SUR (Surface only data)] - Datos tomados usando algún tipo de cubeta, o desde termosatógrafos.
[APB (Autonomous Pinniped Bathythermograph)] - Batitermógrafos sujetos a elefantes marinos. La temperatura y salinidad se registran durante las inmersiones y se transmiten a satélite al salir a la superficie.
[MRB (Moored buoys)] - Plataformas ancladas o estabilizadas para medir datos oceanográficos y atmosféricos en áreas pequeñas alrededor de una ubicación geográfica fija. La temperatura es medida usando termistores. La salinidad es medida usando sensores de conductividad.
[PFL (Profiling floats)] - Dispositivos que se desplazan a una determinada profundidad en la columna de agua, ascendiendo a la superficie en intervalos de tiempo establecidos.
[DRB (Drifting buoys)] - Dispositivos arrastrados por las corrientes oceánicas, ya sea en la superficie o a una determinada profundidad.
[UOR (Undulating Oceanographic Recorders)] - Vehículos oceanográficos remolcados detrás de una embarcación mientras asciende y desciende en la columna de agua.
[GDL (Gliders)] - Vehículos submarinos autónomos reutilizables. Se desplazan desde la superficie hasta una profundidad programada y viceversa.

Un pronóstico es una predicción de algún evento futuro. Es considerado como un problema que abarca muchos campos, incluidos las ciencias ambientales, industria, gobierno, economía, medicina, finanzas, política y ciencias sociales.

Los problemas de pronóstico son frecuentemente clasificados como de corto plazo (días, semanas, meses), mediano plazo (uno o dos años) y largo plazo (varios años). La mayoría de los problemas de pronósticos involucran el uso de series temporales. La razón de que el pronóstico sea tan importante es que la predicción de eventos futuros es una entrada crítica de muchos tipos de procesos de planeación y de toma de decisiones [15].

### **2. 4.1. Modelos estadísticos para series de tiempo**

Los modelos estadísticos brindan las siguientes ventajas: son simples y transparentes, es decir, pueden ser entendidos claramente en términos de sus parámetros. Debido a las expresiones matemáticas simples que los definen, es posible derivar sus propiedades de interés en una forma rigurosamente estadística. Estos modelos pueden ser aplicados a conjuntos de datos bastante pequeños y aún así obtener buenos resultados. Es posible obtener un buen desempeño sin el riesgo del sobre ajuste [17].

Algunos modelos estadísticos para series temporales son: el modelo Auto Regresivo (AR), el modelo de Medias Móviles (MA), el modelo de Medias Móviles Integrado Auto Regresivo (ARIMA), el modelo de Auto Regresión Vectorial (VAR), y los modelos jerárquicos [17].

Existen muchos otros modelos estadísticos desarrollados para series temporales. Algunos de los cuales expanden el modelo ARIMA, mientras que otros hacen diferentes supuestos subyacentes sobre la dinámica temporal que los utilizados en el modelo ARIMA. Algunos ejemplos de estos modelos son: ARIMA Estacional (SARIMA), Heterocedasticidad Condicional Auto regresiva (ARCH), modelos Jerárquicos [17].

### **2. 4.2. Aprendizaje automático y aprendizaje profundo para series temporales**

En los modelos de aprendizaje automático para las series temporales el objetivo es identificar patrones que describan el comportamiento del proceso de forma relevante para predecir el resultado de interés, como la etiqueta de clasificación apropiada para una serie temporal. Algunos ejemplos de estos modelos son: árboles de decisión (bosque aleatorio, árbol con gradiente aumentado), agrupamiento [17].

El aprendizaje profundo es una rama del aprendizaje automático. Aplicado a las series temporales brinda la posibilidad de modelar comportamientos de alta complejidad y no lineales sin tener que adivinar o suponer formas funcionales. En esta técnica, se construye un grafo que conecta nodos de entrada con una complicada estructura de nodos y aristas, donde, al pasar de un nodo a otro a través de una arista, un valor es multiplicado por el peso de la arista y, usualmente, pasado por algún tipo de función de activación no lineal. Es la función de activación no lineal la que hace a la técnica de aprendizaje profundo interesante, ya que permite adaptar

datos altamente complejos y no lineales [17].

## 2. 5. Aprendizaje automático (*Machine Learning*)

El aprendizaje automático es una rama de la inteligencia artificial y de la informática, y se define como: la ciencia (y arte) de programar computadoras para que puedan aprender de los datos [18]. Arthur Samuel, pionero en la investigación en inteligencia artificial, define el aprendizaje automático como: “El campo de estudio que da a las computadoras la habilidad de aprender sin que hayan sido explícitamente programadas para ello”. Tom Mitchell, científico de la informática y pionero del aprendizaje automático, lo define como: “El estudio de algoritmos informáticos que permite a los programas informáticos mejorar automáticamente a través de la experiencia” [18].

Los sistemas basados en aprendizaje automático pueden ser clasificados de acuerdo a las siguientes categorías [18]:

- Si son entrenados o no con supervisión humana, se clasifican como supervisado, no supervisado, semi-supervisado, y aprendizaje reforzado. Dentro de las tareas típicas de este tipo de aprendizaje están la clasificación y la regresión, donde en esta última se trata de predecir un valor numérico objetivo, dado un conjunto de características llamado predictores.
- La forma en la que pueden o no aprender gradualmente: aprendizaje en línea o aprendizaje por lotes.
- Ya sea que funcionen comparando nuevos datos con datos conocidos, o que detecten patrones en los datos de entrenamiento y construir un modelo predictivo: aprendizaje basado en instancias o aprendizaje basado en modelos.

Uno de los retos principales del aprendizaje automático, además de la selección del algoritmo adecuado, es contar con la mayor cantidad de datos disponibles para entrenamiento, ya que el correcto funcionamiento del algoritmo depende de que se cuente con la mayor cantidad de datos, para la mayoría de los modelos de aprendizaje automático las cantidades son de miles de datos [18].

## 2. 5.1. Introducción a las Redes Neuronales Artificiales

El campo de las Redes Neuronales Artificiales (RNA) se ocupa de la investigación de modelos computacionales inspirados por teorías y observación de la estructura y función de las redes biológicas de neuronas en el cerebro. Una RNA se compone generalmente de una colección de neuronas artificiales que están interconectadas para realizar algún cálculo sobre un patrón de entrada y producir un patrón de salida. Son sistemas adaptativos capaces de modificar su estructura interna, típicamente los pesos entre los nodos en la red, lo que les permite ser utilizados para una variedad de problemas de aproximación de funciones como clasificación, regresión, extracción de características y memoria de contenido direccionable [19]. Una de las claves para comprender la RNA es saber que la aplicación del modelo implica que no se esta completamente seguro de las relaciones de los nodos de entradas y los nodos de salidas [20].

La primera arquitectura de red neuronal artificial fue introducida por el neurofisiólogo Warren McCulloch y por el matemático Walter Pitts, en el año 1943. Ellos presentaron un modelo computacional simplificado de como las neuronas biológicas podían trabajar juntas para realizar cálculos complejos usando lógica proposicional. Fue este modelo de las neuronas biológicas el que posteriormente se conoció como una neurona artificial, el cual tenia una o más entradas binarias y una salida binaria. La neurona artificial activa su salida cuando más de cierto número de sus entradas están activas [18].

## 2. 5.2. El Perceptrón

La arquitectura más simple de RNA es conocida como perceptrón, inventada por Frank Roseblatt en 1957. Basada en una neurona artificial llamada unidad lógica de umbral (*Threshold Logical Unit* - TLU) o, también conocida como unidad umbral lineal (*Linear Threshold Unit* - LTU), donde las entradas y las salidas son ahora números (en lugar de valores binarios) y cada conexión de entrada esta asociada con un peso. La TLU, mostrada en la Figura 2.3, calcula una suma ponderada de sus entradas ( $z = w_1x_1 + w_2x_2 + \dots + w_nx_n = x^T w$ ), y posteriormente le aplica una función escalón (*step function*) y proporciona como salida el resultado:  $h_w(x) = \text{step}(z)$ , donde  $z = x^T w$ . Las funciones más comúnmente usadas en el perceptrón son la función escalón unitaria y la función signo. La TLU puede ser usada para clasificación binaria lineal simple, calcula una combinación lineal de sus entradas y si el resultado es mayor a cierto umbral, su

salida es la clase positiva, en caso contrario su salida es la clase negativa.

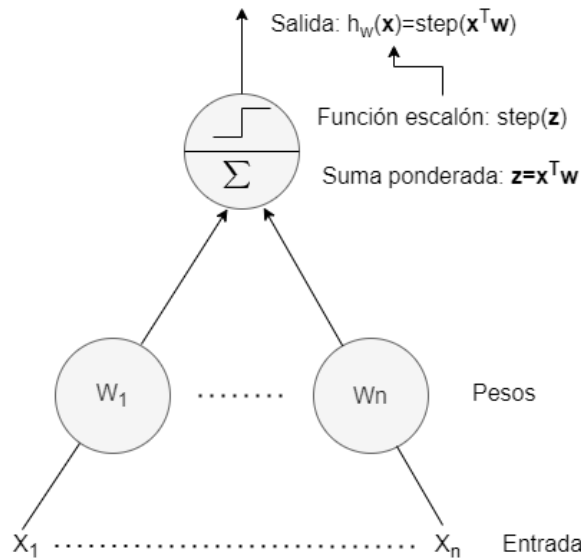


Figura 2.3: Unidad lógica de umbral

Un perceptrón, mostrado en la Figura 2.4, se compone de una capa sencilla de TLU's, con cada TLU conectada con todas las entradas. Cuando todas las neuronas en una capa están conectadas con cada una de las neuronas en la capa previa, la capa se denomina capa totalmente conectada o capa densa. Se agrega una característica extra conocida como sesgo (*bias*) ( $x_0 = 1$ ). Mediante la ecuación 2.4 se representa la salida de una capa totalmente conectada.

$$h_{W,b}(X) = \phi(XW + b) \quad (2.4)$$

Donde:

- $X$  representa la matriz de características de entrada,
- $W$  la matriz de pesos,
- $b$  el vector de sesgo,
- $\phi$  la función de activación.

El perceptrón es entrenado con una variante de la regla de Hebb's que toma en cuenta el error producido por la red; refuerza las conexiones que ayudan a reducir el error. El perceptrón es alimentado con una instancia de entrenamiento a la vez, y para cada instancia hace sus

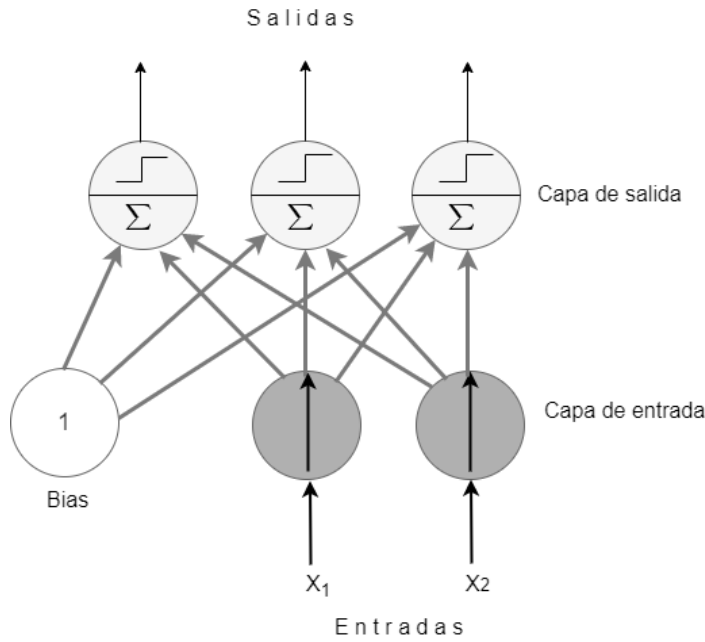


Figura 2.4: Ejemplo de Perceptrón

predicciones. Para cada neurona de salida que produce una predicción errónea, refuerza los pesos de las conexiones que puedan contribuir a la predicción correcta. La ecuación 2.5 representa la regla de aprendizaje o actualización de pesos.

$$w_{i,j}^{(siguientepaso)} = w_{i,j} + \eta(y_j - \hat{y}_j)x_i \quad (2.5)$$

Donde:

- $w_{i,j}$  representa el peso de la conexión entre la  $i^{esima}$  neurona de entrada y la  $j^{esima}$  neurona de salida,
- $x_i$  es la  $i^{esimo}$  valor de entrada de la instancia actual de entrenamiento,
- $\hat{y}_j$  es la salida de la  $j^{esima}$  neurona de salida para la actual instancia de entrenamiento,
- $y_j$  es la salida objetivo de la  $j^{esima}$  neurona de salida,
- $\eta$  es la taza o razón de aprendizaje.

Finalmente, Rosenblatt demostró que si las instancias de entrenamiento son linealmente separables, el perceptrón convergería a una solución, es esto se le conoce como el teorema de convergencia del perceptrón.

### 2. 5.3. Perceptrón multicapa (MLP)

La arquitectura de red neuronal artificial más simple es la conocida como perceptrón multicapa (*multilayer perceptrón*), la cual se muestra en la Figura 2.5. Esta RNA está compuesta de una capa de entrada, una o más capas ocultas, y finalmente una capa de salida. El flujo de información se da solamente en una dirección, desde la entradas a la salida, de ahí que también sea conocida como red neuronal pre-alimentada (*Feed-forward Neural Network*).

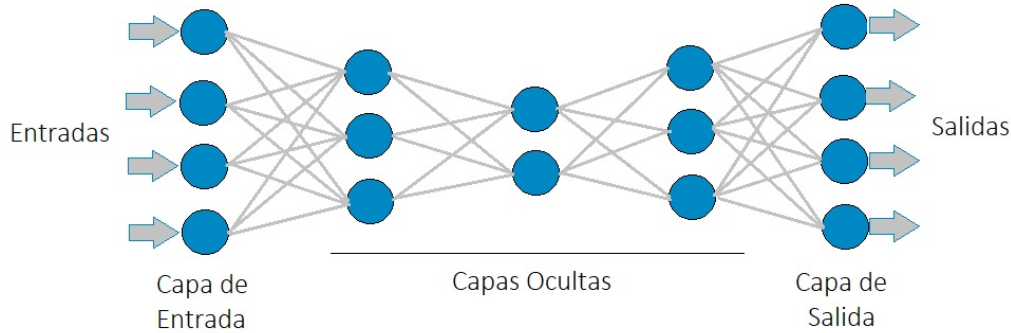


Figura 2.5: Ejemplo de MLP

Por varios años se intentó encontrar una forma de entrenar MLP sin éxito. Fue hasta 1986, cuando Rumelhart, Hinton, y Williams, publicaron el algoritmo de entrenamiento denominado retropropagación (*backpropagation*). Este algoritmo es un método para entrenar los pesos en la RNA. Para cada instancia de entrenamiento, el algoritmo primero hace la predicción (pasada hacia adelante), mide el error, entonces va a través de cada capa en reversa para medir la contribución al error de cada conexión (pasada hacia atrás), y finalmente ajusta los pesos de las conexiones para reducir el error (paso de descenso de gradiente). El proceso se repite hasta que la red converge a la solución. La ecuación 2.6 representa la activación de una neurona dado un patrón de entrada es la siguiente:

$$activation = \left( \sum_{k=1}^n w_k \times x_k \right) + w_{bias} \times 1.0 \quad (2.6)$$

Donde:

- $n$  es el numero de pesos y entradas,
- $x_k$  es la  $k^{esimo}$  atributo del  $i^{esimo}$  patrón de entrada, y
- $w_{bias}$  es el sesgo.

La salida de la neurona se calcula mediante la función de transferencia logística, también conocida como función sigmoidea 2.7.

$$output = \frac{1}{1 + exp(-a)} \quad (2.7)$$

Donde:  $a$  representa la activación de la neurona.

Mediante la ecuación 2.8 se calcula el error para cada neurona de salida.

$$e_i = (c_i - o_i) \times td_i \quad (2.8)$$

Donde:

- $e_i$  es el error para la  $i^{esima}$  neurona,
- $c_i$  es la salida esperada, y
- $o_i$  es la salida actual del  $i^{esima}$  neurona.
- El termino  $td$  es la derivada de la salida de la  $i^{esima}$  neurona.

Para las neuronas ocultas el error se calcula mediante la ecuación

$$e_i = \left( \sum_{k=1}^n (w_{ik} \times es_k) \right) \times td_i \quad (2.9)$$

Donde:

- $e_i$  es el error para la  $i^{esima}$  neurona,
- $w_{ik}$  es le peso entre las neuronas  $i^{esima}$  y  $k^{esima}$ , y
- $es_k$  es el error de la  $k^{esima}$  neurona.

Las derivadas del error para cada peso son calculadas combinando la entrada y el error para cada neurona 2.10.

$$ed_i = \sum_{k=1}^n es_i \times x_k \quad (2.10)$$

Donde:

- $ed_i$  es la derivada del error de la  $i^{esima}$  neurona,



- $es_i$  es el error para la  $i^{esima}$  neurona, y
- $x_k$  es la entrada desde la  $k^{esima}$  neurona en la capa previa.

Finalmente, los pesos son actualizados en la dirección que reduce la derivada del error  $ed_i$ , medido por un coeficiente de aprendizaje 2.11.

$$w_i(t + 1) = w_i(t) + (ed_k \times learn_{rate}) \quad (2.11)$$

Donde:

- $w_i(t + 1)$  es el  $i^{esimo}$  peso actualizado,
- $ed_k$  es la derivada del error para la  $k^{esima}$  neurona, y
- $learn_{rate}$  parámetro coeficiente de actualización.

Aunque actualmente los problemas de análisis de series de tiempo son realizados con redes neuronales recurrentes, así como también con redes neuronales convolucionales, existen varias razones por las cuales considerar a está arquitectura: son altamente paralelizables, lo que las hace muy eficaces. Son una buena prueba para determinar si realmente existen dinámicas complejas en el eje del tiempo en la secuencia. Frecuentemente se utiliza esta arquitectura como parte de un modelo más complejo [17].

Una variación de la arquitectura de red neuronal pre-alimentada, como la ilustrada en la Figura 2.6, que podría mejorar su desempeño con datos secuenciales consiste en implementar un mecanismo de atención en la red. Este mecanismo de atención le permite a la red neuronal aprender en que parte de la secuencia centrarse y que parte de la secuencia de entrada podría relacionarse con los resultados deseados. Esto se realiza a través de pesos de atención, los cuales se ajustan a cada paso de tiempo para que el modelo aprenda cómo combinar información de diferentes pasos de tiempo. Los pesos de atención multiplican lo que sería la salida o el estado oculto de un modelo, convirtiéndolo en un vector de contexto, llamado así ya que el estado oculto esta ahora mejor contextualizado para tener en cuenta y relacionar toda la información contenida en la serie de tiempo a lo largo del tiempo [17].

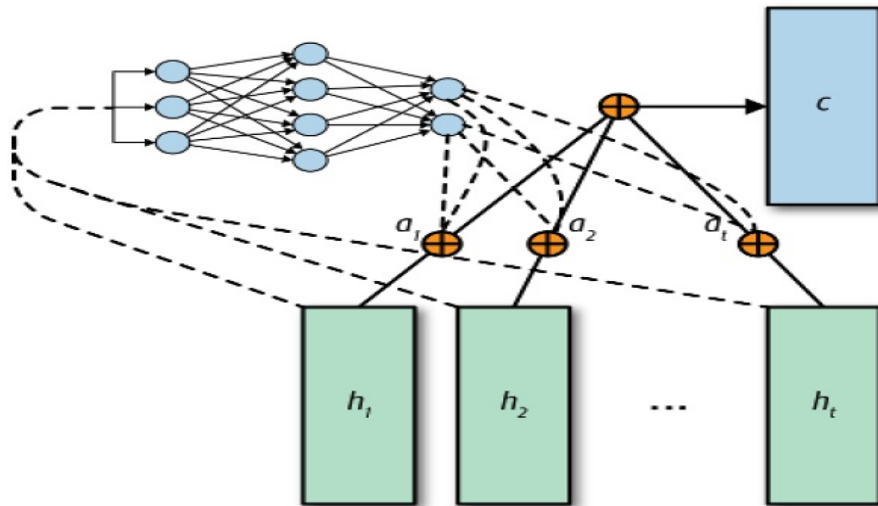


Figura 2.6: MLP con Mecanismo de Atención

## 2. 5.4. Función de activación

Un cambio clave en la arquitectura MLP fue la función de activación, en lugar de utilizar la función escalón se utilizó la función logística 2.12, ya que tiene una derivada bien definida diferente de cero en todas partes, permitiendo al descenso de gradiente tener cierto progreso en cada paso. El algoritmo de retropropagación funciona bien con otras funciones de activación además de la función logística, dos de las más populares son:

- Tangente hiperbólica 2.13: al igual que la función logística, tiene forma de  $S$ , es continua, y diferenciable, pero sus valores de salida están en el rango de -1 a 1. Esta función frecuentemente ayuda a acelerar la convergencia.
- Unidad lineal rectificada 2.14, la cual es continua pero no diferenciable en  $z = 0$ , y su derivada es 0 para  $z < 0$ . Sin embargo, en la práctica funciona muy bien y tiene la ventaja de ser rápida de calcular.

$$\sigma(z) = 1/(1 + \exp(-z)) \quad (2.12)$$

$$\tanh(z) = 2\sigma(2z) - 1 \quad (2.13)$$

$$\text{ReLU}(z) = \max(0, z) \quad (2.14)$$

## 2. 5.5. Algoritmos optimizadores

Además del descenso de gradiente, existen otros algoritmos o métodos usados para cambiar los atributos de la RNA, como los pesos y la tasa de aprendizaje, para reducir el error o pérdida. Estos algoritmos son conocidos como optimizadores o de optimización y son responsables de reducir las pérdidas y proporcionar el resultado más exacto posible. El descenso de gradiente es el algoritmo de optimización más básico pero más ampliamente utilizado. Es un algoritmo de primer orden el cual depende de la derivada de primer orden de la función de pérdida.

Otro algoritmo de optimización es Adam, que significa estimación de momento adaptativo (*adaptive moment estimation*). Este algoritmo realiza un seguimiento de un promedio exponencialmente decreciente de gradientes pasados, así como también una pista de un promedio exponencialmente decreciente de gradientes cuadrados pasados. Debido a que Adam es un algoritmo de tasa de aprendizaje adaptativo, requiere menos ajuste de la tasa de aprendizaje ( $\eta$ ), lo que hace que sea aún más fácil de usar que el descenso de gradiente. Trabaja con momentos de primero y segundo orden. Una de sus principales ventajas es que es muy rápido y converge rápidamente, aunque, es computacionalmente costoso.

## 2. 5.6. Redes Neuronales Recurrentes (RNN)

Las RNN son una amplia clase de redes neuronales donde los mismos parámetros son aplicados una y otra vez incluso cuando las entradas cambian a medida que pasa el tiempo. Entre sus principales características están: Una RNN ve los pasos de tiempo uno a la vez y en orden. Tiene un estado que preserva de un paso de tiempo a otro, y con este estado, así como con sus parámetros estadísticos, determina su respuesta a actualizaciones de información en cada paso de tiempo. Tiene parámetros que la ayudan a actualizar su estado, así como su estado oculto, en cada paso de tiempo [17].

En el artículo [21] se define a las RNN como máquinas de aprendizaje que calculan recursivamente nuevos estados aplicando funciones de transferencia a estados y entradas previas. Las funciones de transferencia típicas están compuestas por una función de transformación afín seguida por una función no lineal, la cual se selecciona dependiendo de la naturaleza del problema en particular. En el contexto de la predicción, una RNN es entrenada con datos temporales de entrada  $x(t)$  para producir una salida temporal deseada  $y(t)$ . La salida  $y(t)$  puede ser cualquier

serie de tiempo relativa a la entrada e inclusive un cambio temporal de la entrada misma  $x(t)$ .

En la Figura 2.7 se puede observar la versión más básica de la arquitectura de una RNN, también conocida como red neuronal recurrente Elman (ERNN), o *Simple RNN* o *Vanilla RNN*, la cual ha sido aplicada en diferentes contextos: como aplicaciones de procesamiento de lenguaje natural, pronóstico de llegada de turistas, predicción de series de tiempo de carga eléctrica, por citar algunos.

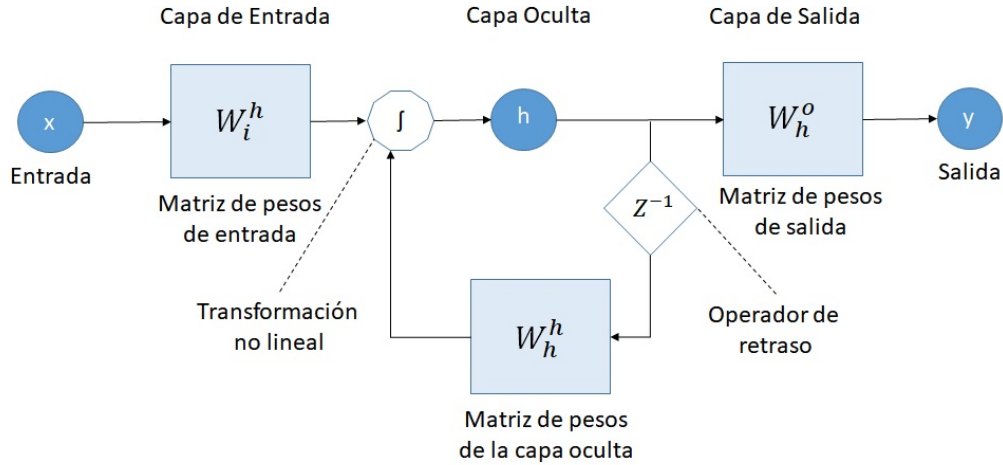


Figura 2.7: Arquitectura simple de RNN

Otra forma de representar una RNN se puede ver en la Figura 2.8, la cual muestra el paradigma de “desenrollado” (*unrolling*), el cual es usado frecuentemente para representar una RNN, debido a que su arquitectura es basada en células.

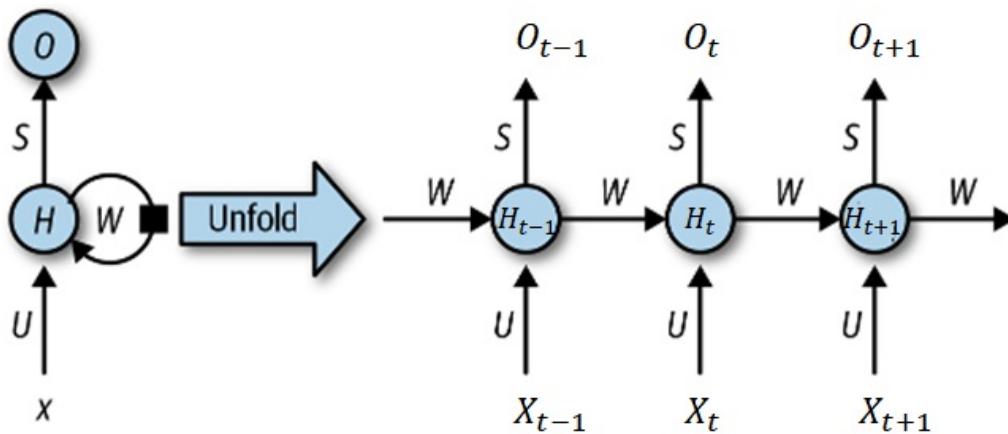


Figura 2.8: “Unrolling” de una RNN

Otra arquitectura de RNN es la conocida como Memoria Grande de Corto Plazo (LSTM). En

esta arquitectura se intenta resolver el problema de desvanecimiento del gradiente no imponiendo ningún sesgo a las observaciones recientes, pero mantiene el error constante fluyendo hacia atrás en el tiempo. A diferencia de la red ERNN, implementa una unidad de procesamiento interna más elaborada llamada celda. Una celda LSTM está compuesta de 5 diferentes componentes no lineales. El estado interno de la celda es modificado por la LSTM a través de interacciones lineales. Lo que permite la propagación hacia atrás de la información suavemente a través del tiempo, con la consecuente mejora de la capacidad de memoria de la celda. La LSTM protege y controla la información en la celda a través de tres compuertas, las cuales son implementadas por una multiplicación sigmoidea y puntual. Para controlar el comportamiento de cada compuerta, se entrena un conjunto de parámetros con gradiente de descenso, para resolver una tarea objetivo [21].

Otra variante de RNN es la arquitectura conocida como Unidad Recurrente Cerrada (GRU), la cual de forma adaptativa captura dependencias en diferentes escalas de tiempo. En una GRU, las compuertas de entrada y olvido se combinan en una sola compuerta de actualización, que controla de forma adaptativa cuánto puede olvidar o recordar cada unidad oculta. El estado interno en la GRU está siempre totalmente expuesto en la salida, debido a la pérdida de mecanismo de control [21].

Otro modelo basado en RNN es el conocido como modelo *autoencoder* o *seq2seq*, el cual es útil para tareas de series de tiempo, como la predicción de series de tiempo univariante o multivariante. Este modelo está formado por dos capas recurrentes, donde la primera capa se ejecuta hasta completarse. Su estado oculto es pasado a la segunda capa, la cual toma este estado oculto y sus propias salidas como nuevas entradas para el siguiente paso.

## 2. 5.7. Redes Neuronales Convolucionales (CNN)

La inspiración para el desarrollo de esta arquitectura surge de los experimentos realizados por David H. Hubel y Torsten Wiesel en gatos, y posteriormente en monos, proporcionando conocimientos cruciales sobre la corteza visual. Ellos mostraron que muchas neuronas en la corteza visual tienen un pequeño campo receptivo local, lo cual significa que reaccionan solamente a estímulos localizados en una limitada región del campo visual. El campo receptivo de diferentes neuronas se puede traslapar, y juntas pueden cubrir todo el campo visual. Además, Mostraron que algunas neuronas reaccionan solamente a imágenes de líneas horizontales, mientras que otras

reaccionan a líneas con diferente orientación. También notaron que algunas neuronas tienen un campo receptivo más grande, y que reaccionan a patrones más complejos que son combinaciones de patrones de más bajo nivel. De estas observaciones surgió la idea de que las neuronas de más alto nivel están basadas en las salidas que producen las neuronas vecinas de más bajo nivel [18].

Estos estudios sobre la corteza visual inspiraron el desarrollo del *neocognitron*, en 1980, que gradualmente evolucionó a lo que actualmente conocemos como red neuronal convolucional, cuya arquitectura típica se muestra en la Figura 2.9. Un hito importante se dio en una publicación en 1998, en la cual se introdujo la arquitectura *LeNet-5*, ampliamente utilizada para reconocimiento de números escritos a mano. Esta arquitectura, a demás de tener fundamentos ya conocidos como las capas totalmente conectadas y las funciones de activación Sigmoides, introdujeron dos nuevos fundamentos: las capas convolucionales y las capas de agrupación (*pooling layers*) [18].

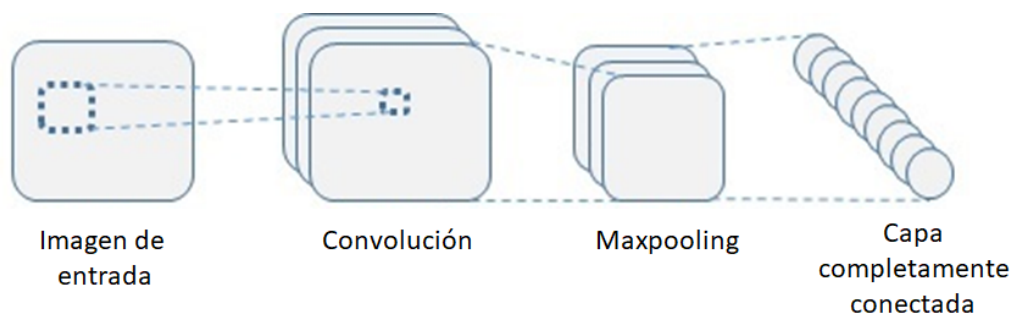


Figura 2.9: Arquitectura típica de una CNN

Como se muestra en la Figura 2.9, la red neuronal convolucional se compone básicamente de lo siguiente [18]:

- 1 Capa de convolución: Se realiza la extracción de características de la imagen de entrada, o de la salida de la capa previa, aplicando filtros para generar mapas de características. Es el componente principal de la *CNN*. Las neuronas en la capa convolucional se conectan sólo con ciertas neuronas en la capa previa. Una neurona localizada en el renglón  $i$ , columna  $j$  de una capa dada está conectada a las salidas de las neuronas en la capa previa localizadas en los renglones  $i$  al  $i + f_h - 1$ , columnas  $j$  a  $j + f_w - 1$ , donde  $f_h$  y  $f_w$  representan lo alto y ancho del campo receptivo. Para que una capa tenga la misma altura y ancho que la capa previa, es común agregar ceros alrededor de la entrada, esto se conoce como relleno con ceros (*zero padding*).

Es posible, también, conectar una entrada más grande a una capa mucho más pequeña espaciando los campos receptivos. El cambio de un campo receptivo al siguiente se conoce como *stride*. En la Figura 2.10 se observa una capa de entrada de  $5 \times 7$  (más *zero padding*) conectada a una capa  $3 \times 4$ , usando un campo receptivo de  $3 \times 3$  con un *stride* de 2. Una neurona localizada en el renglón  $i$ , columna  $j$  en la capa 2 está conectada a la salida de las neuronas en la capa 1 (capa previa) localizadas en los renglones  $ixs_h$  a  $ixs_h + f_h - 1$ , columnas  $jx_s_w$  a  $jx_s_w + f_w - 1$ , donde  $s_h$  y  $s_w$  son los *strides* vertical y horizontal.

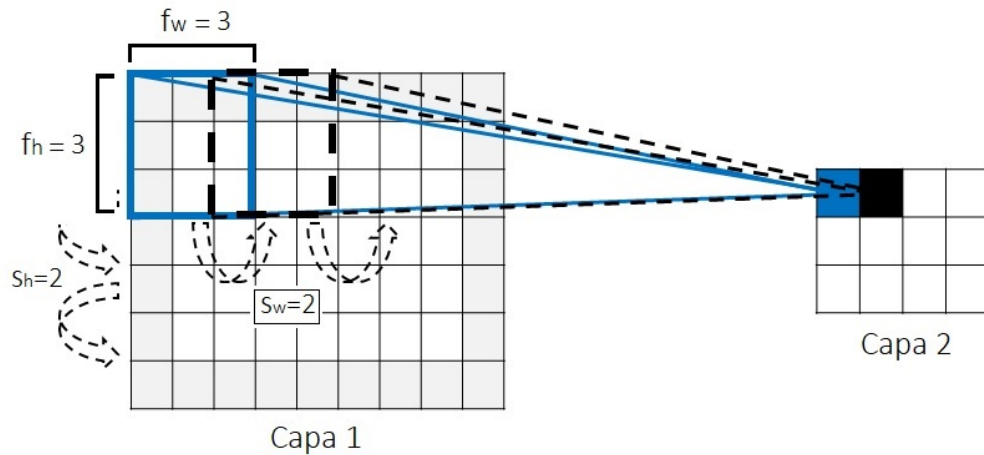


Figura 2.10: Reduciendo la dimensionalidad usando un *stride* de 2

2 Filtros: Los pesos de las neuronas se pueden representar como una imagen pequeña del tamaño del campo receptivo. Este conjunto de pesos se conoce como filtro o núcleos de convolución (*convolution kernels*).

Durante el proceso de convolución ocurre una reducción de la dimensionalidad en cada neurona de la capa oculta, como se muestra en la Figura 2.10:

- Toma entradas desde el filtro.
- Calcula la suma de los pesos.
- Aplica el *bias*.

En la Figura 2.11 se muestra como se aplica a la imagen de entrada un filtro  $3 \times 3$  (matriz de pesos  $w_{i,j}$ ). El cálculo del valor para cada neurona de la capa oculta se obtiene con la fórmula 2.15 y se aplica una función no-lineal, como por ejemplo la función *ReLU* 2.14,

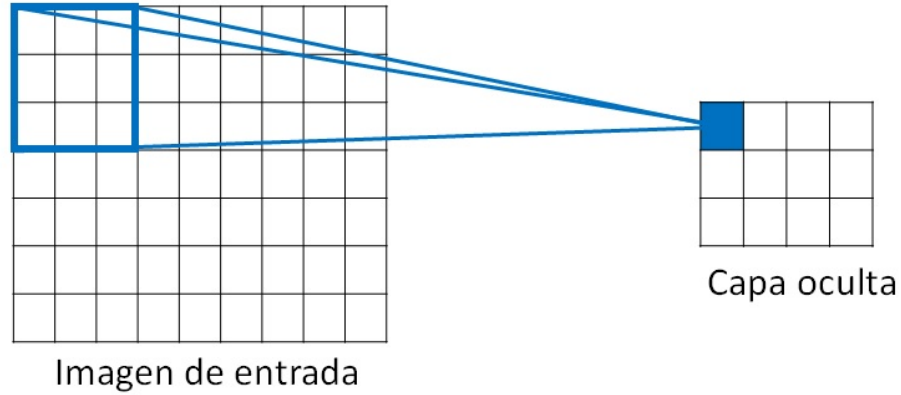


Figura 2.11: Proceso de convolución

la cual reemplaza todos los valores negativos por cero.

$$z = \sum_{i=1}^3 \sum_{j=1}^3 w_{i,j} x_{i+p,j+q} + b \quad (2.15)$$

3 Múltiples mapas de características, mostrada en la Figura 2.12: Lo más común en una *CNN* es que una capa convolucional tenga múltiples filtros, y que genere un mapa de características por filtro.

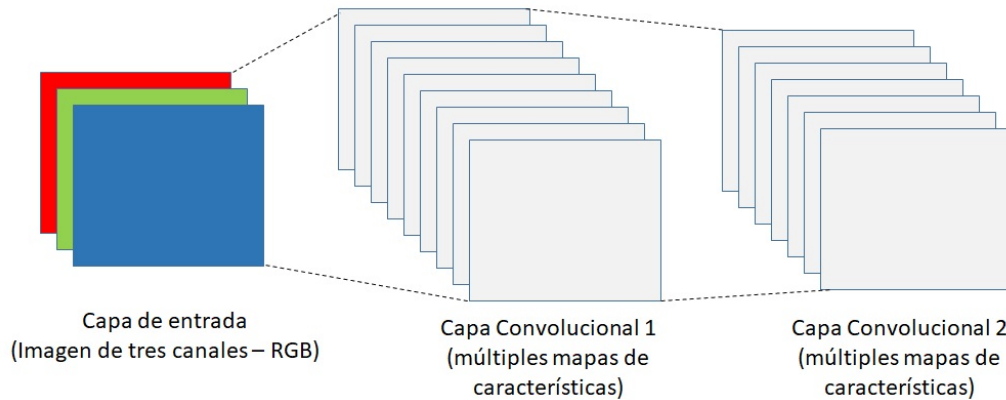


Figura 2.12: Capas de convolución con múltiples mapas de características

4 Capa de agrupación (*pooling layer*), la cual se puede ver en la Figura 2.13: El objetivo de esta capa es reducir la imagen de entrada para reducir la carga computacional, la cantidad de memoria usada, y el número de parámetros.



Así como en las capas de convolución, cada neurona en la capa de agrupación esta conectada a las salidas de un número limitado de neuronas en la capa previa, localizadas con un campo receptivo rectangular pequeño. Las neuronas de agrupación no tienen pesos, lo que ellas hacen es agregar las entradas usando una función de agregación como el máximo o la media.

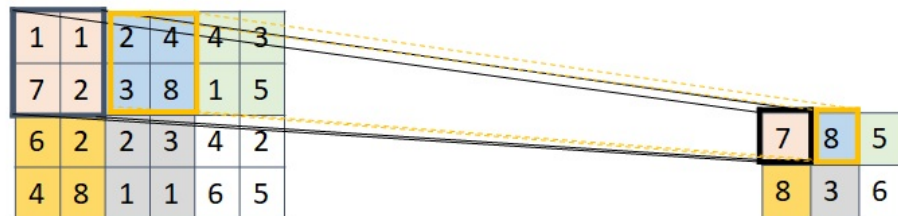


Figura 2.13: Agrupación, aplicando un filtro de  $2 \times 2$ , y un *stride* de 2

La principal característica de la convolución es que todos los espacios se tratan por igual, lo que tiene sentido para las imágenes pero no para las series de tiempo, donde algunos puntos en el tiempo están necesariamente más cerca que otros. Por lo anterior, las redes neuronales convolucionales son usadas más frecuentemente como componente de una red para análisis de series de tiempo.

Algunos usos de las redes convolucionales para aplicaciones de series de tiempo incluyen: establecen un patrón de navegación de un usuario de internet, lo que ayuda a detectar actividad de navegación anormal. Identificación de patrones de latidos cardíacos anómalos a partir de datos electrocardiogramas. Generar predicciones de tráfico basadas en grabaciones pasadas de múltiples ubicaciones en una ciudad [17].

## 2. 5.8. Arquitecturas combinadas

Muy a menudo, en la industria como en concursos de aprendizaje profundo, surgen arquitecturas novedosas para pronósticos de series de tiempo. Tal es el caso de la arquitectura propuesta por investigadores de la *Carnegie Mellon University*, quienes explotaron las fortalezas de las arquitecturas de redes convolucionales y de las redes recurrentes, agregando otras innovaciones como la capa denominada “*skip recurrent*” mediante la cual el modelo se podía ajustar para

poner atención a las periodicidades (anual, semanal, diaria) presente en los datos. El modelo fue adaptado usando un modelo lineal tradicional conocido como modelo autoregresivo.

El modelo propuesto, denominado LSTNet modificado, fue la suma de las salidas del modelo autoregresivo y un modelo construido con una capa recurrente y una capa *skip recurrent* en paralelo. Las entradas de cada capa recurrente fueron las salidas de las capas convolucionales. En la Figura 2.14 podemos ver la arquitectura del modelo propuesto:

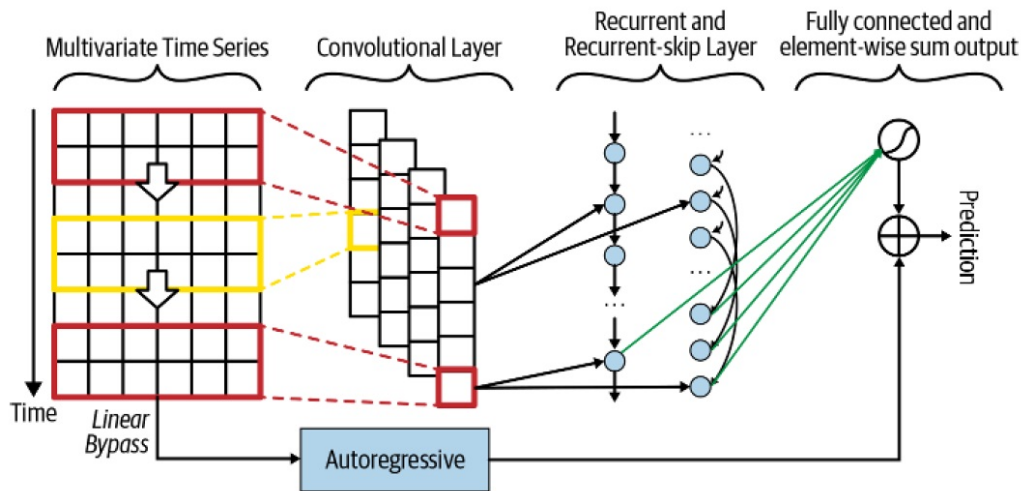


Figura 2.14: Arquitectura del modelo LSTNet modificado

En el artículo [22] se presentan los resultados, hallazgo y conclusiones del concurso denominado *M4 competition*. Esta competencia es la continuación de tres competencias previas iniciadas hace 45 años cuyo propósito era aprender como mejorar la exactitud de los pronósticos, y cómo se puede aplicar ese aprendizaje para avanzar en la teoría y práctica de la predicción.

Entre los principales hallazgos y conclusiones de la competencia M4, están los siguientes:

- De los 17 métodos más exactos, 12 fueron combinaciones de enfoques principalmente estadísticos.
- Un modelo híbrido compuesto por características del enfoque estadístico y de aprendizaje automático. Este modelo produjo los pronósticos e intervalos de predicción más precisos.
- El segundo método más preciso fue una combinación de siete métodos estadísticos y uno de aprendizaje automático, con los pesos para el promedio siendo calculados por un algoritmo de aprendizaje automático que fue entrenado para minimizar el error de pronóstico a través de pruebas de exclusión.

- Los seis métodos puros de aprendizaje automático presentados en la competición tuvieron un desempeño pobre.

En conclusión, la precisión de métodos estadísticos o de aprendizaje automático individuales es baja, y los modelos híbridos o aquellos formados por combinaciones de métodos son el camino a seguir para mejorar la precisión de los pronósticos.

## 2. 6. Antecedentes del pronóstico de series temporales mediante aprendizaje automático

Esta sección muestra un resumen de artículos consultados donde se ha abordado el pronóstico de series temporales con aprendizaje automático. Uno de los artículos es [23], donde se analizan los diseños de codificadores y decodificadores comunes utilizados en el pronóstico de series temporales, describiendo como la información temporal es incorporada en las predicciones por cada modelo. Se estudian las redes neuronales convolucionales, redes neuronales recurrentes, redes recurrentes de memoria de corto plazo (LSTM). También se destacan los desarrollos recientes en modelos híbridos, los cuales combinan modelos estadísticos con redes neuronales.

En [17] el autor establece que, en los modelos de aprendizaje automático para las series temporales el objetivo es identificar patrones que describan el comportamiento del proceso de forma relevante para predecir el resultado de interés, además de brindar la posibilidad de modelar comportamientos de alta complejidad y no lineales sin tener que adivinar o suponer formas funcionales.

Otro estudio donde se aborda el aprendizaje automático para series temporales es en [24], donde, se analizan arquitecturas de aprendizaje profundo de redes neuronales artificiales como la red neuronal perceptrón multicapa, algunos modelos de redes neuronales recurrentes, así como redes neuronales convolucionales. También se incluye información referente a la implementación de cada una de las arquitecturas, librerías para optimización de hiper-parámetros. Se analiza, también, el desempeño del *hardware*.

En el artículo [25], se hace una evaluación de modelos estadísticos, de aprendizaje profundo, y de aprendizaje automático, para predecir los parámetros hidrográficos: temperatura superficial del mar (*SST*) y altura de ola significativa (*SWH*). Se comparan los modelos predictivos

con cuatro diferentes métricas de evaluación: el error medio absoluto (MAE), el error cuadrático medio (MSE), la raíz del error cuadrático medio (RMSE), y el coeficiente de determinación R cuadrada ( $R^2$ ). Los modelos evaluados fueron: el modelo estadístico ARIMA, los modelos de aprendizaje automático regresión lineal, regresión por soporte vectorial, árbol de decisión, y bosque aleatorio; y finalmente, el modelo de aprendizaje profundo propuesto es un modelo híbrido formado por capas apiladas de redes neuronales recurrente de tipo unidad de compuerta cerrada (GRU) y redes neuronales perceptrón multicapa. Los resultados experimentales mostraron que el modelo de aprendizaje profundo superó ligeramente a los modelos de aprendizaje automático en el rendimiento general, y ambos superaron al modelo estadístico.

En lo que respecta al uso de redes neuronales artificiales para la predicción y pronóstico de datos hidrográficos, existen trabajos de investigación como [26] donde estudia la limitación de las redes neuronales artificiales (RNA) de ser específicas del sitio y no proporcionar información espacial simultánea similar a los esquemas numéricos, presentando predicciones a escala de cuenca de temperatura superficial del mar (SST) basadas en la operación de RNA. Se usó una red neuronal con arquitectura FFBP (feed-forward back-propagation) para predecir valores mensuales de anomalías de temperatura superficial del mar (SSTA). El número de neuronas de la capa oculta se determinó con pruebas o ensayos. La capa de salida con 1 neurona. Para la capa oculta se usó la función de transferencia *tan-sigmoid*, y en la capa de salida la función *linear*. El conjunto de datos fue dividido en subconjuntos de entrenamiento, validación cruzada, y prueba, en proporciones de 75 %, 5 %, y 20 %, respectivamente. Los datos de SST de la región del Océano Índico Tropical fueron extraídos de la información global HadISST (Hadley Centre Sea Ice and SST dataset). Los valores de SST son especificados en cada nodo de una cuadrícula de  $1^\circ \times 1^\circ$ , con un total de 3813 nodos. Se configuró un modelo de RNA para cada uno de nueve meses. Por lo tanto, hubo un total de  $3813 \times 9$  RNA. Cada una fue entrenada con 60 combinaciones de entrada. Durante el entrenamiento y para cada combinación de entradas, se usaron 10 diferentes condiciones de iniciales aleatorias. El algoritmo de optimización utilizado fue *Levenberg-Marquardt*. El objetivo del entrenamiento fue minimizar el error cuadrático medio. Se conservó el mejor modelo resultante de las 10 diferentes condiciones iniciales y 60 diferentes condiciones de entrada. Las métricas para la evaluación de desempeño del modelo fueron el coeficiente de correlación  $r$  y la raíz del error cuadrático medio normalizada (NRMSE). En la predicción a un mes el valor obtenido de  $r$  fue superior a 0.8 para la mayor parte del área de

estudio. Sin embargo, la exactitud decrece para 2 o 3 meses, mostrando un valor de  $r$  mayor de 0.6.

En el artículo [27], se presenta un modelo de RNA (Red Neuronal Artificial) para predecir la temperatura superficial del mar (SST) y delinear los frentes de la SST en el noreste del mar de Arabia. El modelo predice un día en el futuro, usa la SST del día actual para predecir la SST del siguiente día. La RNA utilizada es de tipo pre-alimentada (FFNN) con algoritmo de retro-propagación *Quasi-Newton*. La capa de entrada consta de 8 neuronas, mientras que la capa oculta contiene 5 neuronas, finalmente la capa de salida con una neurona. El desempeño del modelo es evaluado por dos criterios estadísticos de error: RMSE y MAE. Para evaluar cuantitativamente el desempeño del modelo en el dominio espacial y temporal y verificar la exactitud de la predicción, se analizó el RMSE para todo el dominio espacial para cada día. El rango de error varía entre 0.2 y  $1.0C$  para la mayoría de los días. También se observó que, más del 80 % (2014 y 2015), 60 % (2013) de las predicciones del modelo tienen un error  $\leq \pm 0.5C$ . Para todos los años más del 90 %, el error es  $\leq \pm 1.0C$  y los errores de aproximadamente  $2.0C$  se presentan en el 1 % de las predicciones. Por último, se realizó un análisis de errores mensuales utilizando el error mensual absoluto. Durante los meses de monzón los errores  $\leq 0.5C$  ocurrieron entre 40-50 % de las predicciones y los errores en el rango de  $0.5 - 1.0C$  entre el 30-40 %, mientras que en los meses de no-monzón los errores  $\leq 0.5C$  ocurrieron entre 70-80 % de las predicciones y los errores en el rango de  $0.5 - 1.0C$  entre el 10-20 %.

Además de los anteriores, existen otros artículos donde se reporta la implementación de RNA diferentes al perceptrón multicapa o pre-alimenta, como lo son las redes neuronales recurrentes (RNN), así como dos de sus variantes más populares y eficientes, las de memoria de largo y corto plazo (LSTM) y unidades recurrentes cerradas (GRU) estudiadas en [28], [29], [30]. En [29] se propone un modelo para predecir la estructura termohalina sub-superficial utilizando datos remotos, mientras que en [30] se presenta un modelo para predecir y estimar datos faltantes en series temporales de océanos costeros.

Las redes neuronales convolucionales (CNN) no han quedado al margen de la predicción y pronóstico de datos de series temporales. Como se puede constatar en [31], donde se utiliza esta RNA para predecir temperaturas sub-superficiales en el océano Pacífico mediante datos superficiales, y en [32] donde utilizan una red neuronal profunda convolucional para reconstruir en alta resolución anomalías de temperatura y salinidad de la superficie e interior del océano.

Referente a modelos híbridos, en [33] se realiza la pronóstico de temperatura basado en datos de series temporales implementando un nuevo modelo de RNA denominado red neuronal recurrente convolucional (CRN). El modelo consiste de una red neuronal convolucional (CNN) y una red neuronal recurrente (RNN), donde la CNN es usada para procesar la correlación espacial en cada mapa de datos de temperatura, mientras que la RNN es usada para procesar la correlación de tiempo. Otro artículo donde se implementa un modelo híbrido es [34], en el cual se propone un método basado en algoritmos de descomposición de modo empírico (EMD) y redes neuronales de retro-propagación (BPNN) para la predicción de la temperatura superficial del mar.

# Capítulo 3

## Metodología

### 3. 1. Selección de datos

#### 3. 1.1. Datos Argo

Para este trabajo se usaron los datos Argo de la base de datos de fecha 10 de junio de 2021, con número de identificación 85023 [35]. Una vez obtenida la base de datos Argo, se descomprime en los directorios `dac` y `geo`. En el directorio `dac`, uno de los archivos descomprimidos es “`ar_index_global_prof.txt.gz`” el cual describe todos los archivos de perfiles contenidos en la base de datos descargada. Se seleccionan los archivos de perfiles de acuerdo al área de interés proporcionando océano, latitud y longitud para un polígono rectangular dado. Se implementó un *script* en lenguaje *Python* mediante el cual se extrajeron los parámetros de fecha, latitud, longitud, presión, temperatura y salinidad, de los archivos de datos en modo diferido identificados por la variable `DATA_MODE` igual a ‘D’, . Los datos extraídos fueron grabados en un nuevo archivo con formato de valores separados por comas (`csv`). De esta forma se crearon algunos subconjuntos de datos para diferentes polígonos para el posterior análisis de los datos.

#### 3. 1.2. Datos de la base de datos oceánica mundial (WOD)

La selección y descarga de los conjuntos de datos de la WOD se realizó del sitio de internet [36]. La búsqueda se realizó proporcionando las coordenadas geográficas del área de interés, las variables a extraer (temperatura, salinidad y presión), así como todos los conjuntos de datos. De esta forma se obtuvo un conjunto de datos, en formato `NetCDF`, para cada una de las áreas

de interés. En la Figura 3.1 se puede observar la cantidad de mediciones totales obtenidas en cada uno de los años en el periodo de 2013 a 2021.

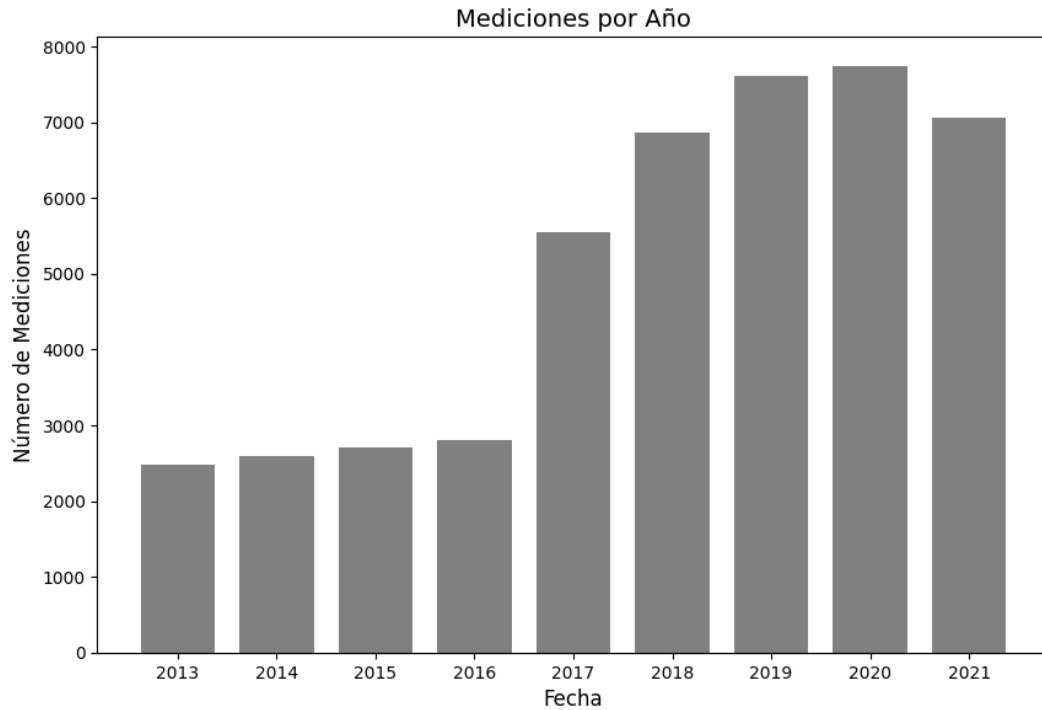


Figura 3.1: Histograma de las mediciones por año

Una vez descargados los conjuntos de datos, se descomprimieron y posteriormente, mediante un script en lenguaje Python, se extrajeron datos de fecha, latitud, longitud, presión, temperatura y salinidad. El script recibe como parámetros la ruta del archivo en formato NetCDF, los vértices  $x$  e  $y$ , y la cantidad de vértices, de los polígonos que delimitan las áreas de interés. Los datos en los cuales las variables *Temperature\_WODprofileflag* y *Salinity\_WODprofileflag* tiene un valor de cero correspondiente a datos aceptados [5] sirven como entrada del algoritmo punto en polígono, si el algoritmo arroja como resultado que el punto se encuentra dentro del polígono, los datos son almacenados en una lista y posteriormente grabados en un archivo con formato de valores separado por comas (csv). De esta forma, al finalizar el procesamiento, se tiene, para cada polígono de interés, un archivo con los datos (registros) que están dentro.



## 3. 2. Análisis exploratorio de datos

Como parte de este análisis se generaron gráficas de dispersión, diagramas de cajas (figuras 4.1 a 4.5). Estas gráficas ayudan a conocer las características del conjunto de datos, a identificar valores faltantes y atípicos, conocer también la distribución espacial y temporal de los datos, así como a identificar patrones como estacionalidad y variabilidad.

## 3. 3. Área de estudio

De los antecedentes de este trabajo se conoce que existen áreas de océano con relativa falta de mediciones *in situ*, y se señalan, específicamente, la zona central y sur del Golfo de México, y la Zona Económica Exclusiva de México, respectivamente.

Por lo anterior, para seleccionar el área de estudio, primero se definieron cuatro áreas delimitadas por polígonos irregulares, tres de las áreas en las costas del Pacífico Mexicano y una en el Golfo de México. A cada una de las áreas propuestas se les realizó el cálculo del perímetro, área y densidad de mediciones contenidas en cada una. Siendo el parámetros de densidad de mediciones el tomado como criterio para la selección del área de estudio. Estas áreas se pueden ver en la Figura 3.2. Las áreas definidas se identificaron como se muestra en la Tabla 3.1:

Tabla 3.1: Áreas propuestas

Área	Descripción	Identificador
I	Costa del Pacífico de la península de Baja California	Area I
II	Costa oeste y sur del Estado de Baja California Sur	Area II
III	Costa del Pacífico sur de México	Area III
IV	Golfo de México	Area IV

## 3. 4. Modelos para predicción espacial

Con el objetivo de completar las series temporales se realiza la predicción espacial, la cual consiste en estimar valores de temperatura en nuevas ubicaciones geográficas, en puntos dentro de los polígonos convexos (ver apéndice A) formados por los datos *in situ* disponibles para cada

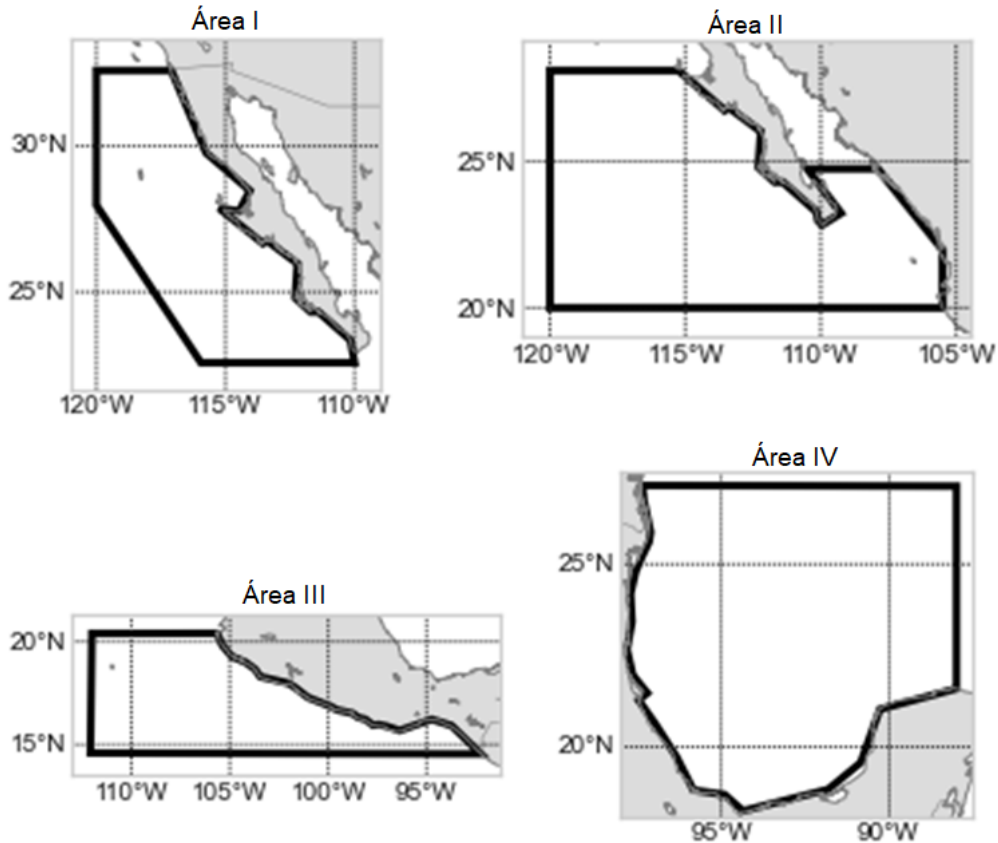


Figura 3.2: Áreas propuestas

mes del período comprendido entre los años 2013 a 2021. Se propone utilizar modelos de red neuronal artificial del tipo perceptrón multicapa o pre-alimentada. Cada modelo se implementó con dos neuronas en la capa de entrada, una neurona en la capa de salida; el número de capas ocultas y el número de neuronas por capa oculta se determinaron de forma experimental.

### 3. 4.1. Selección del número de capas ocultas y neuronas por capa oculta

Se realizó un experimento utilizando la librería Python para optimizar hiperparámetros de nombre Talos [37]. En este experimento se hizo variar el número de capas ocultas, así como el número de neuronas de cada capa oculta. Se seleccionó el modelo que proporcionó el menor valor de la función de pérdida ( $MSE$ ) con los datos de validación. Se utilizaron como datos de entrada para este experimento las mediciones del mes de octubre del año 2019, al ser el mes

con la mayor cantidad de mediciones. Mediante el experimento se determinó que la RNA para la predicción de temperatura tendría cinco capas ocultas, con 200 neuronas en cada capa.

### 3. 4.2. Implementación de los modelos de RNA

Los modelos de RNA fueron creados utilizando el modulo *keras* de la librería de Python *TensorFlow*. Se utilizó el modelo *Sequential* para crear la RNA, la cual está compuesta de capas apiladas conectadas secuencialmente. Todas las capas utilizan la función de activación ReLU, con excepción de la capa de salida que no utiliza función de activación y donde se tiene una sola neurona debido a que se predice un solo valor (temperatura). A continuación se muestra el código para la implementación del modelo de RNA para la predicción espacial de temperatura.

```
model = keras.Sequential(  
    [layers.Dense(200, activation='relu',  
input_shape=[2]),  
    layers.Dense(200, activation='relu'),  
    layers.Dense(200, activation='relu'),  
    layers.Dense(200, activation='relu'),  
    layers.Dense(1)])
```

Una vez creado el modelo, se compila utilizando el método *compile* del modelo *Sequential* para especificar la función de pérdida y optimizador utilizados, así como la métrica para evaluación del entrenamiento.

```
model.compile(  
    loss='mse',  
    optimizer=tf.keras.optimizers.Adam(  
learning_rate=0.0001),  
    metrics=[tfa.metrics.RSquare()])
```

Con el argumento *callbacks.ModelCheckpoint* y el parámetro *save\_best\_only=True* es posible grabar en un archivo con extensión *.h5* la arquitectura del modelo, incluyendo los hiperparámetros de cada capa, como los pesos y los valores de sesgo. También se graban los hiperparámetros y estados de la función de optimización. En el modelo propuesto, se graban los hiperparámetros cuando el desempeño del modelo con el conjunto de datos de validación sea el mejor.

```
checkpoint_cb =
```

```
keras.callbacks.ModelCheckpoint(  
    filename+".h5", save_best_only=True)
```

El argumento *callbacks.EarlyStopping* permite interrumpir el entrenamiento cuando el valor de la métrica de desempeño no muestre progreso o mejora con los datos de validación por un número de épocas determinado por el parámetro *patience*.

```
early_stopping_cb = keras.callbacks.EarlyStopping(  
    patience=20, restore_best_weights=True)
```

Finalmente el modelo es entrenado utilizando el método *fit*, pasando como argumentos las entradas (x) y valores objetivo (y) de los datos de entrenamiento y validación. Otro argumento del método es el número de épocas, el cual tiene un valor de cinco mil. El método devuelve un objeto *History* el cual contiene los parámetros de entrenamiento, épocas, y un diccionario con los valores obtenidos de la función de pérdida y la métrica de desempeño medidos al final de cada época con el conjunto de datos de entrenamiento y con el conjunto de datos de validación.

```
history = model.fit(  
    x_scaled, y_scaled, epochs=5000,  
    validation_data=(val_x_scaled, val_y_scaled),  
    callbacks=[checkpoint_cb, early_stopping_cb] )
```

### 3. 4.3. Selección de los datos de entrenamiento, validación, y prueba

El conjunto de datos filtrado fue dividido utilizando la función *train\_test\_split* de la librería *scikit-learn* de Python. Realizando división aleatoria del conjunto de datos con la proporción 70/20/10% de los datos para entrenamiento, validación, y prueba, respectivamente. La división de los datos se efectuó para cada mes a nivel de celda de la cuadrícula de  $2^{\circ}x2^{\circ}$  creada. El procedimiento efectuado fue: para cada mes, recorrer cada una de las celdas con mediciones, dividir el conjunto de mediciones, y al terminar de recorrer las celdas obtener un conjunto de datos para las entradas (x) y valores objetivo (y) para los datos de entrenamiento, validación, y prueba, respectivamente. Estos conjuntos de datos se almacenaron en estructuras de datos *DataFrame* de la librería *Pandas* de Python.

El paso siguiente fue el escalamiento de los datos, utilizando la función *StandardScaler* de la librería *scikit-learn* de Python. Esta función permite estandarizar los datos eliminando la media

y escalando los datos de forma que su varianza sea igual a uno. Los valores estandarizados son calculados por la fórmula 3.1.

$$x_{hat} = (x - \mu)/\sigma \quad (3.1)$$

Donde:  $\mu$  es la media de los datos de entrenamiento,  $\sigma$  es la desviación estándar de los datos de entrenamiento, y  $x$  es el valor a estandarizar.

### 3. 4.4. Entrenamiento de los modelos de RNA

Se entrenó un modelo de RNA para cada mes en el periodo comprendido del año 2013 al año 2021, es decir, se utilizó una RNA para las mediciones disponibles en cada mes (mediciones de cada mes en apéndice A). Con la finalidad de obtener el modelo con el menor valor de la función de pérdida con los datos de prueba, el entrenamiento se repitió diez veces para cada mes, esto debido a que la división de los datos para obtener los conjunto de datos de entrenamiento, validación, y prueba, así como la inicialización de los pesos, se realiza de forma aleatoria, variando los resultados del entrenamiento con cada división obtenida.

### 3. 4.5. Evaluación de los modelos de RNA

Los modelos de red neuronal artificial, una vez entrenados, se utilizaron para realizar la predicción espacial en ubicaciones dentro de los polígonos convexos formados por las mediciones de cada mes en los años comprendidos entre 2013 a 2021. Se generaron ubicaciones cada  $0.1^\circ$  tanto de latitud como de longitud. Con esto se obtuvieron valores de temperatura en nuevas ubicaciones.

Los modelos de red neuronal artificial utilizados para la predicción espacial fueron evaluados con los datos de prueba mediante las siguientes métricas: error cuadrático medio (MSE), raíz cuadrada del error cuadrático medio (RMSE), error absoluto medio (MAE), y con el coeficiente de determinación ( $R^2$ ), las cuales son utilizadas para la evaluación de modelos para realizar regresión en diversos trabajos de investigación.

### 3. 4.6. Predicción espacial en nuevas ubicaciones

Después de haber efectuado el entrenamiento de cada uno de los modelos de RNA, se realiza la predicción en nuevos puntos (ubicaciones) para cada mes del periodo de estudio. El primer paso realizado fue generar estos nuevos puntos dentro del polígono convexo formado por las mediciones disponibles para cada mes. Se tomaron ubicaciones con separación  $0.1^\circ$ , tanto en latitud como en longitud. Posteriormente, la latitud y longitud de cada ubicación fue normalizada utilizando la técnica de estandarización, donde los valores son centrados alrededor de la media y con una unidad de desviación estándar, la formula utilizada fue 3.2.

$$\hat{x} = \frac{x - \mu}{\sigma} \tag{3.2}$$

Donde:  $\mu$  es la media de los valores de entrada, y  $\sigma$  es la desviación estándar.

# Capítulo 4

## Resultados

### 4. 1. Conjuntos de datos

Como parte de los resultados del presente trabajo, se muestran a continuación las gráficas correspondientes al análisis exploratorio de los datos del área de estudio seleccionada.

La primer gráfica, Figura 4.1, muestra las mediciones en tres rangos de profundidad, en la Figura 3.6 (a) se muestran las mediciones de cero a diez metros, en la Figura 3.6 (b) las mediciones correspondientes al rango de 45 a 55 metros, y por último, en la Figura 3.6 (c) las mediciones en el rango de 95 a 105 metros.

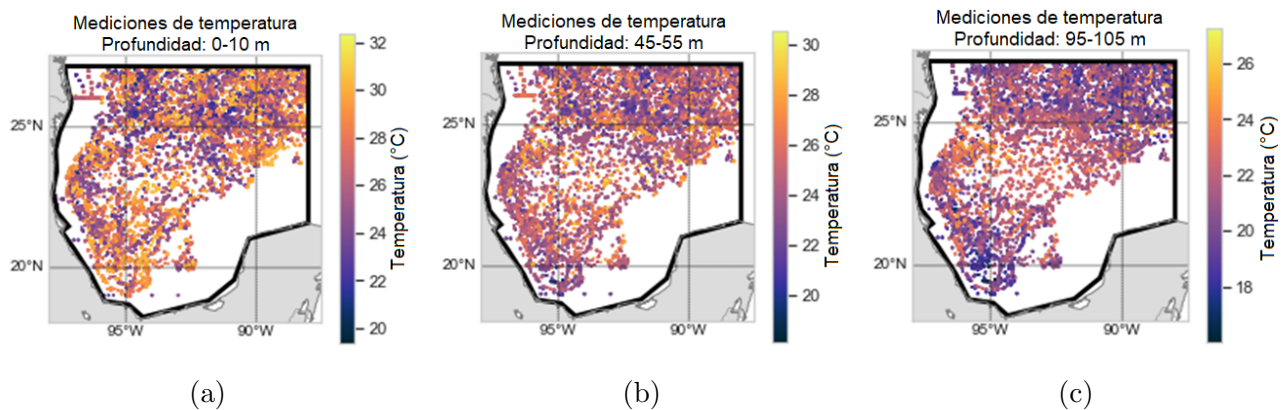


Figura 4.1: Gráficas de dispersión de temperatura

La siguiente gráfica, Figura 4.2, ilustra las mediciones de profundidad (superior en color azul), temperatura (al centro en color rojo), y salinidad práctica (inferior en color verde), en el periodo de tiempo comprendido entre los años 2010 al 2021. Esta gráfica ayuda a visualizar

la continuidad de las mediciones en el tiempo, así como el rango de valores que alcanza cada parámetro.

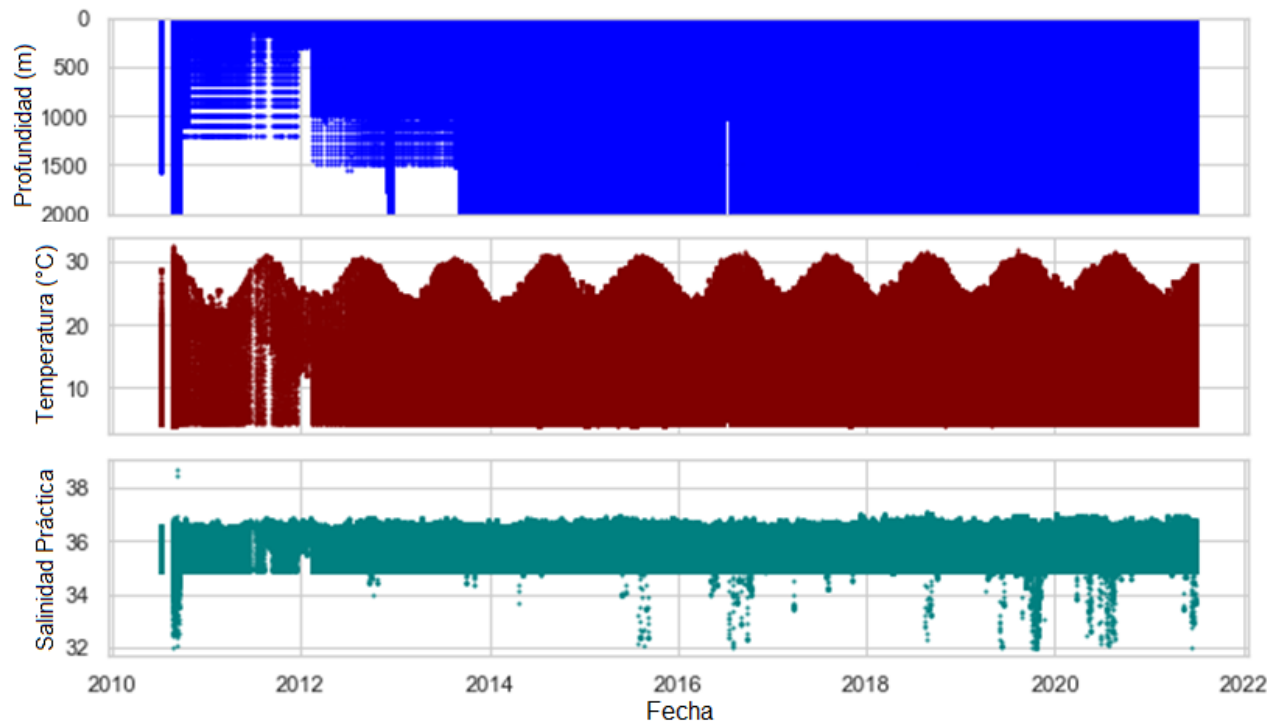


Figura 4.2: Gráficas de dispersión de presión, temperatura y salinidad, respecto al tiempo

La gráfica de cajas y bigotes (*boxplot*) de la Figura 4.3 muestra la distribución de las mediciones mensuales de temperatura en tres rangos de profundidad. La gráfica superior en el rango de cero a diez metros, la gráfica intermedia en el rango de cuarenta y cinco a cincuenta y cinco metros, finalmente, la gráfica inferior en el rango de noventa y cinco a ciento cinco metros. En esta gráfica se puede identificar la concentración de valores de temperatura en los tres rangos de profundidad, y también es posible observar la variabilidad a través del tiempo.

Otra gráfica generada, es la gráfica de dispersión de la Figura 4.4, la cual también muestra la temperatura en tres rangos de profundidad, pero en el periodo comprendido entre los años 2013 a 2021. En esta gráfica podemos ver la variabilidad interanual, así como la estacionalidad que muestra este parámetro.



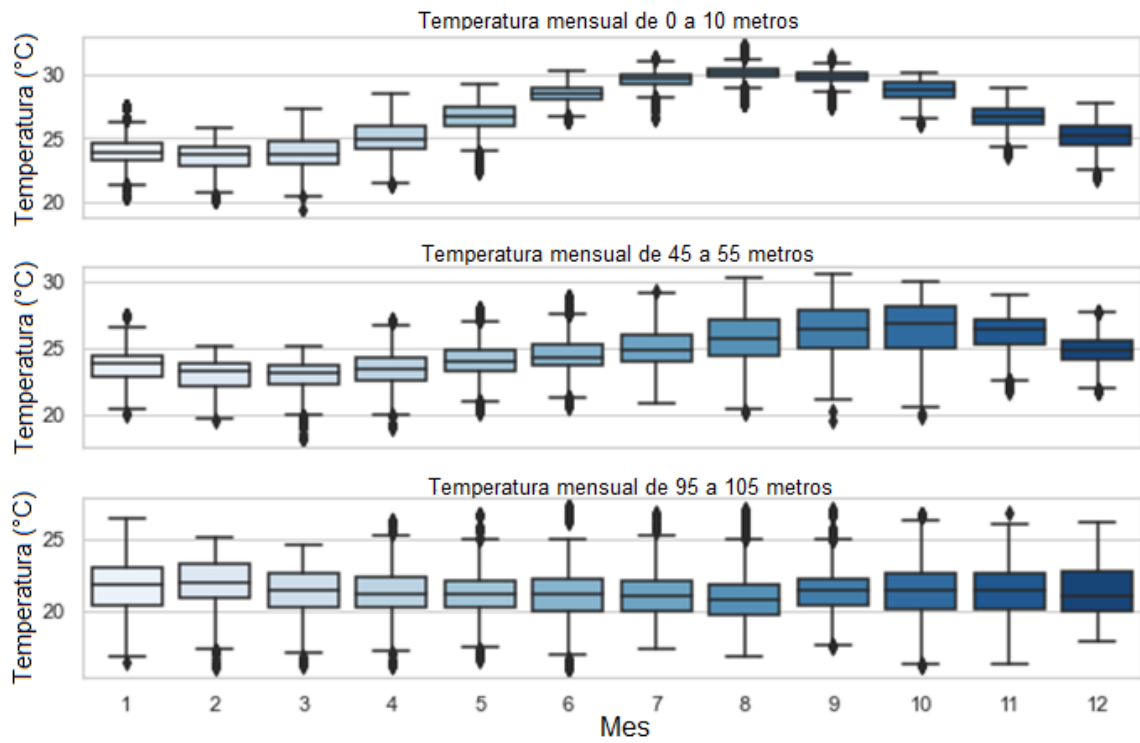


Figura 4.3: Gráficas de cajas de temperatura mensual

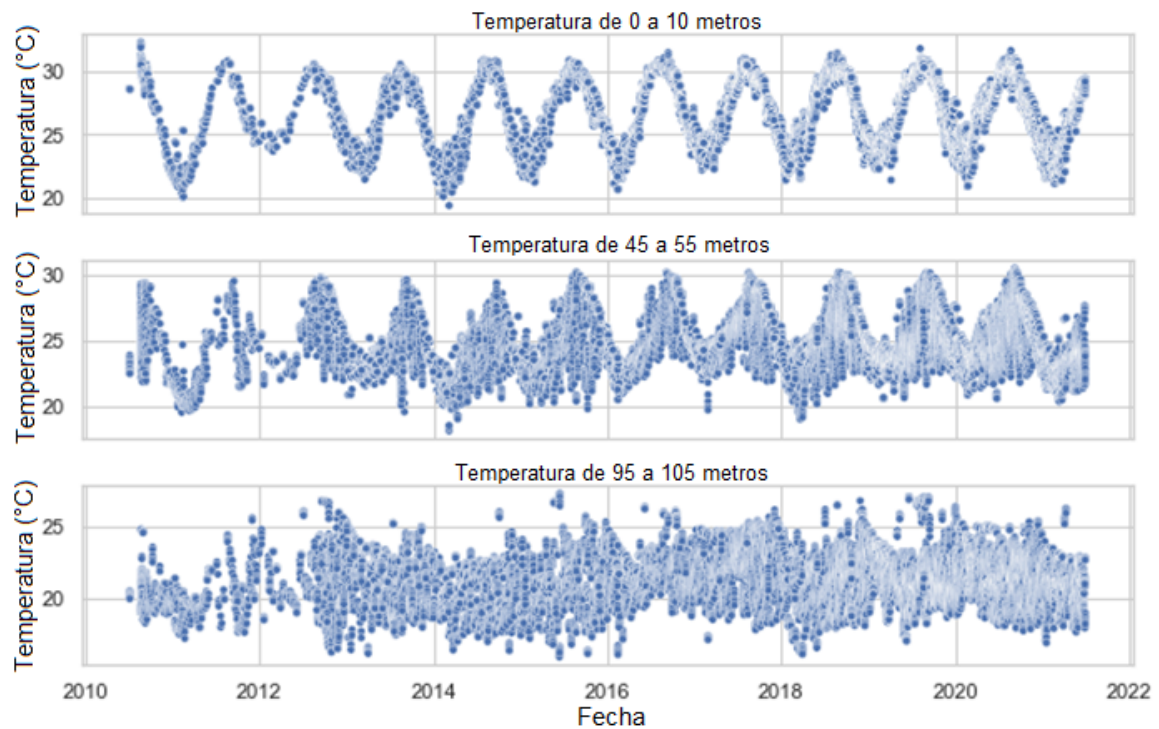
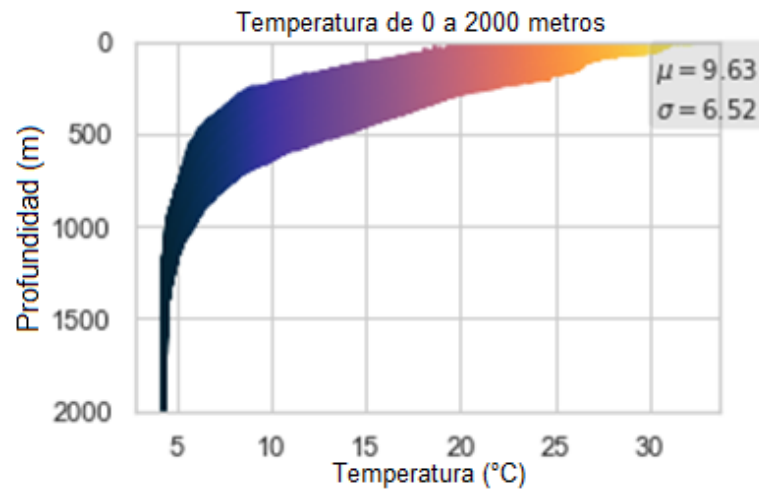
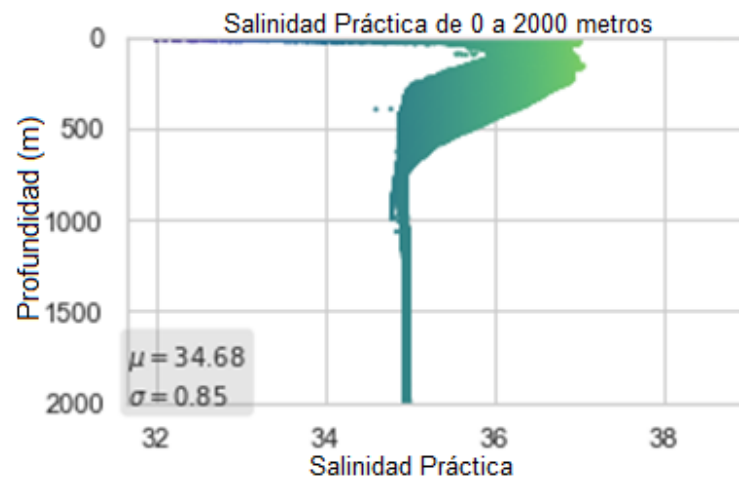


Figura 4.4: Gráficas de dispersión de temperatura mensual

La Figura 4.5 muestra gráficas de dispersión, donde es posible ver el rango de valores de cada parámetro desde cero a dos mil metros de profundidad. La Figura 4.5 (a) muestra los valores de temperatura en el eje horizontal y la profundidad en el eje vertical. La Figura 4.5 muestra la salinidad práctica en el eje horizontal y la profundidad en el eje vertical.



(a) Gráficas de dispersión temperatura



(b) Gráficas de dispersión salinidad

Figura 4.5: Gráficas de dispersión de temperatura y salinidad práctica

Posterior al análisis exploratorio de los datos, se propusieron cuatro áreas delimitadas por polígonos irregulares, para seleccionar de entre estas el área de estudio. Se calcularon el perímetro, área y densidad de mediciones para cada una de las áreas propuestas. En la Tabla 4.1 se puede ver el perímetro resultante para cada una de las áreas. La Tabla 4.2 muestra el área resultante.

Tabla 4.1: Perímetro calculado para cada área

Área	Perímetro calculado ( $Km$ )
I	3,715.39
II	4,900.31
III	5,123.89
IV	3,550.42

Tabla 4.2: Área calculada para cada área propuesta

Área	Área calculada ( $Km^2$ )
I	550,405.96
II	954,776.31
III	813,279.70
IV	802,043.34

La Figura 4.6 muestra la distribución espacial de las mediciones disponibles en cada área propuesta, mientras que la Tabla 4.3 muestra la densidad de mediciones de cada una. Finalmente, tomando como criterio la densidad de mediciones por kilómetro cuadrado, se seleccionó como área de estudio para el presente trabajo de investigación, el área IV (Golfo de México), la cual se muestra en la Figura 4.7.

Tabla 4.3: Densidad de mediciones de cada área propuesta

Área	Perímetro ( $Km$ )	Área ( $Km^2$ )	Densidad ( $Observaciones/Km^2$ )
I	3,751.39	550,405.96	2.35
II	4,900.31	954,776.31	1.68
III	5,123.89	813,279.70	1.15
<b>IV</b>	3,550.42	802,043.34	<b>4.09</b>

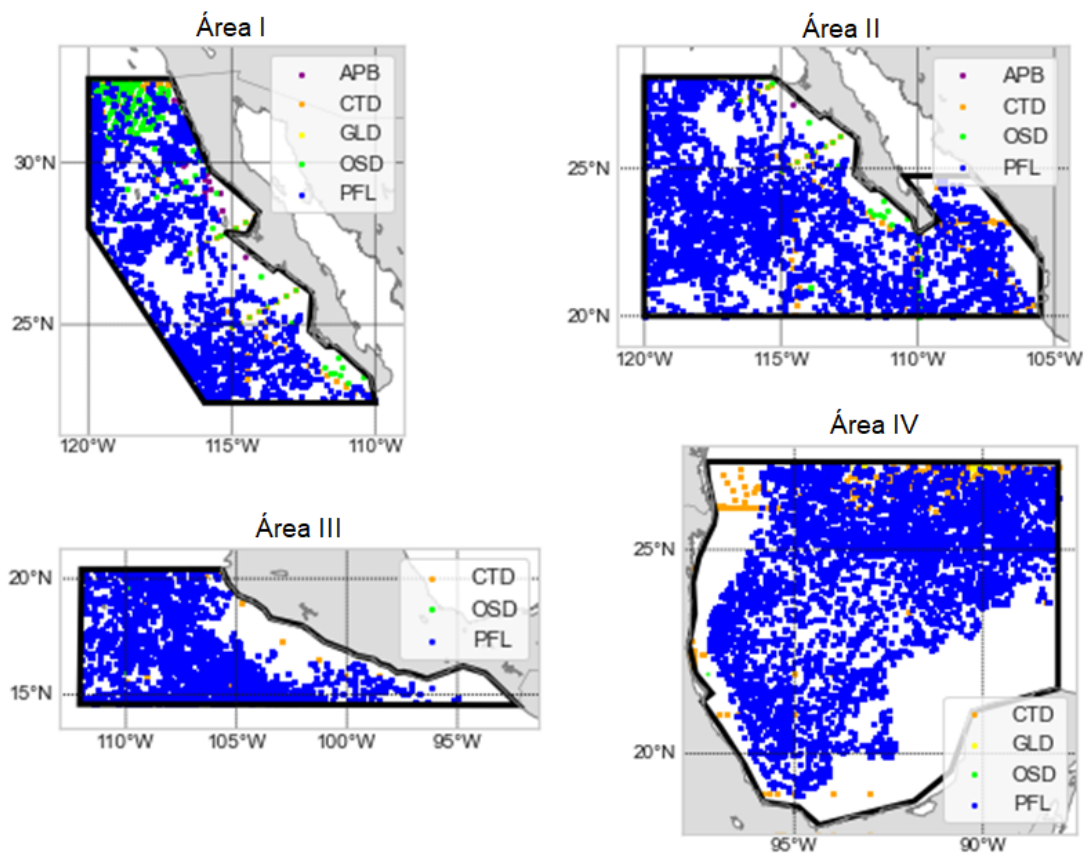


Figura 4.6: Distribución de mediciones de cada área propuesta

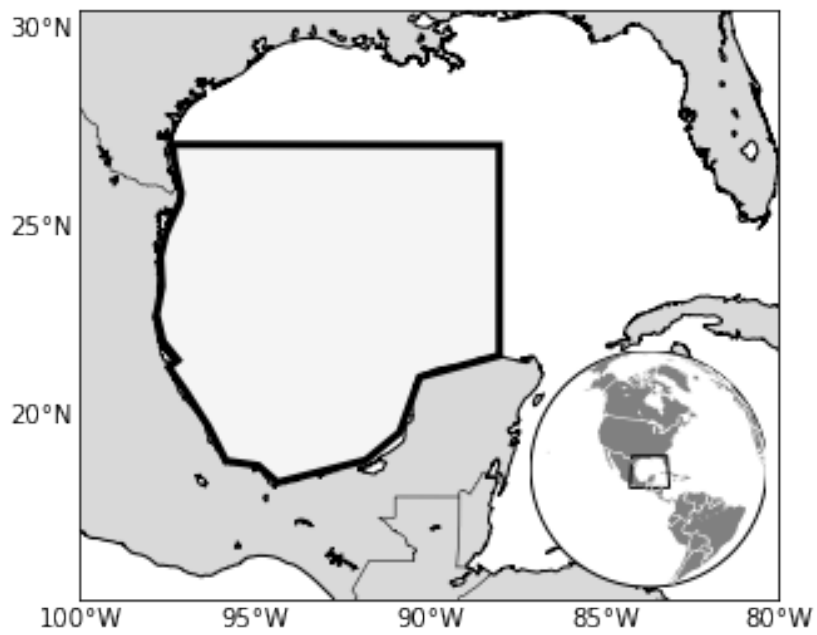


Figura 4.7: Área de estudio seleccionada

## 4. 2. Modelo de Red Neuronal Artificial

Para la predicción espacial se utilizó un modelo de red neuronal artificial pre-alimentada también conocida como perceptrón multicapa, mostrada en la Figura 4.8, donde podemos ver las dos neuronas en la capa entrada, una neurona en la capa de salida, y de forma abreviada las cuatro capas ocultas con 100 neuronas por capa.

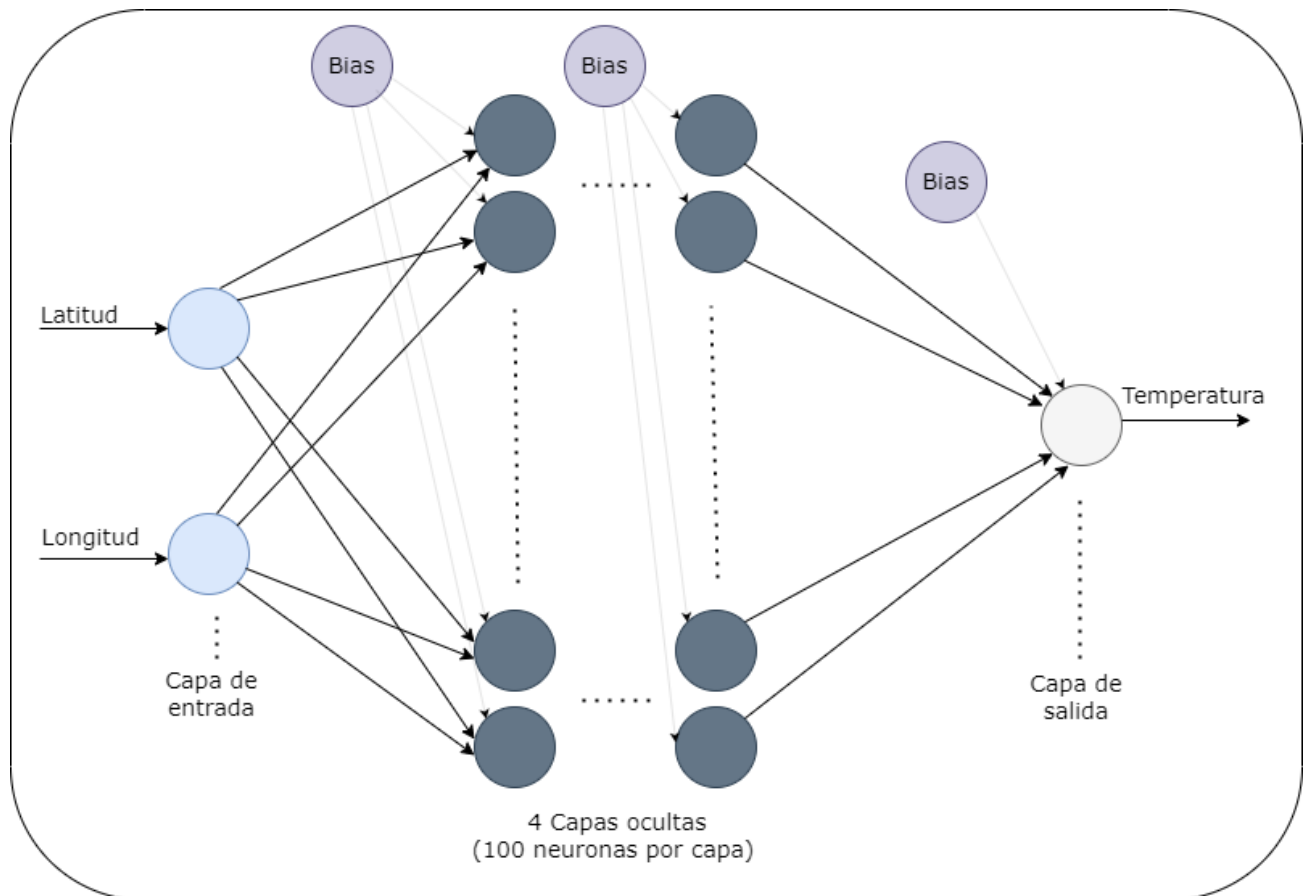


Figura 4.8: Red Neuronal Pre-alimentada

Los hiperparámetros seleccionados para los modelos de predicción espacial se muestran en la Tabla 4.4.

## 4. 3. Resultados del entrenamiento

El conjunto de datos utilizado para el entrenamiento de los modelos de RNA, formado por las mediciones mensuales disponibles en el rango de profundidad de cero a diez metros

Tabla 4.4: Hiperparámetros de los modelos de RNA para predicción espacial

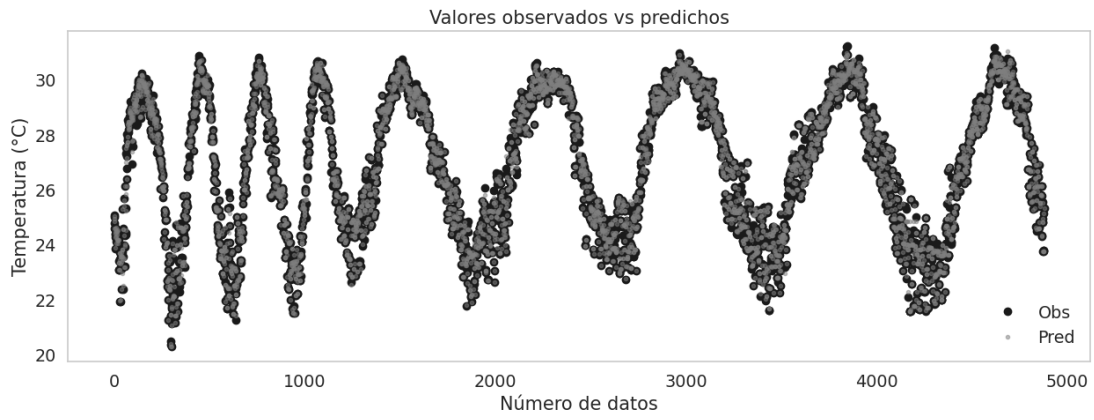
Hiperparámetro	Descripción
Función de pérdida	Error cuadrático medio ( $MSE$ )
Función de activación	Unidad lineal rectificadora ( $ReLU$ )
Optimizador	Adam
Tasa de aprendizaje	0.0001
Métricas de desempeño	Coefficiente de determinación ( $R^2$ )
Número máximo de épocas	5000
Criterio de parada	<i>Earlystopping</i>
Paciencia	100

(mostrado en el apéndice A), fue dividido, de forma aleatoria, en tres subconjuntos de datos: de entrenamiento, validación, y prueba, en la proporción de 70 %, 20 %, y 10 %, respectivamente.

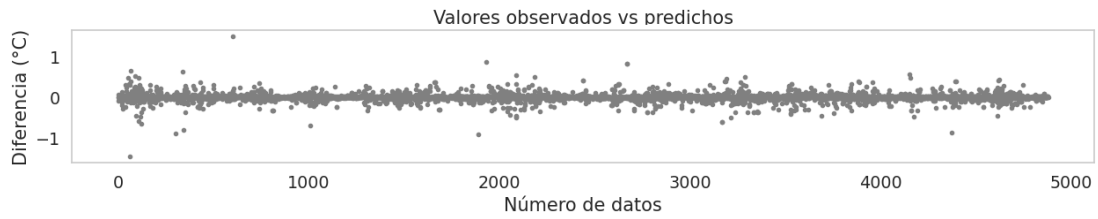
En la figura 4.9 se muestran los valores de temperatura predichos por los modelos de RNA comparados contra las observaciones *in situ*. Se puede observar, en la imagen (a), que los valores predichos muestran una gran similitud con las observaciones. En la imagen (b) se puede observar que la diferencia (error) entre los valores observados y los valores predichos por la RNA se encuentran agrupados al centro.

La gráfica de dispersión es otra herramienta que permite visualizar la relación lineal entre dos variables. En la figura 4.10 se muestra la gráfica de dispersión entre las observaciones y las predicciones. Se puede apreciar que la mayoría se encuentran agrupadas sobre la línea diagonal, lo que indica que las predicciones y las observaciones tienen valores muy similares.

Otra forma de visualizar la distribución de la diferencia entre las observaciones y las predicciones es el diagrama de cajas y bigotes. En la figura 4.11 se pueden ver los diagramas de cajas y bigotes para la distribución del error absoluto (la magnitud de la diferencia) para cada uno de los doce meses del año. En la figura 4.11(a) (con valores atípicos), es posible observar que solo dos puntos tienen un error absoluto mayor a  $1.0^{\circ}\text{C}$ , mientras que en la figura 4.11(b) (sin valores atípicos) se puede observar que el 75 % del error absoluto muestran un valor inferior a  $0.10^{\circ}\text{C}$ .



(a)



(b)

Figura 4.9: Gráfica de dispersión de temperaturas observadas vs predichas

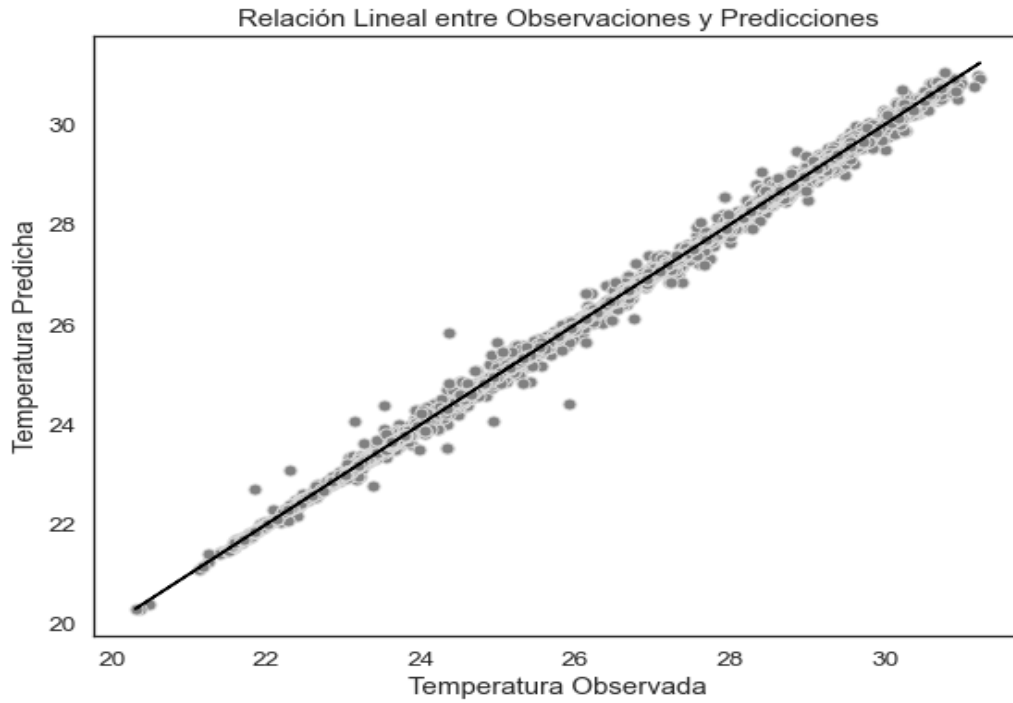
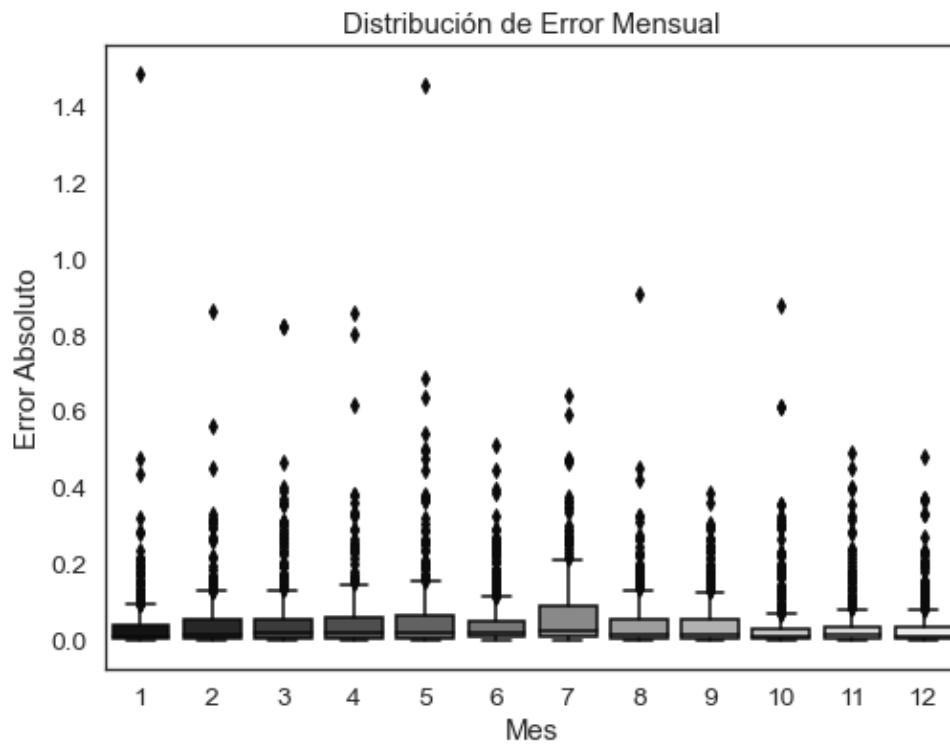
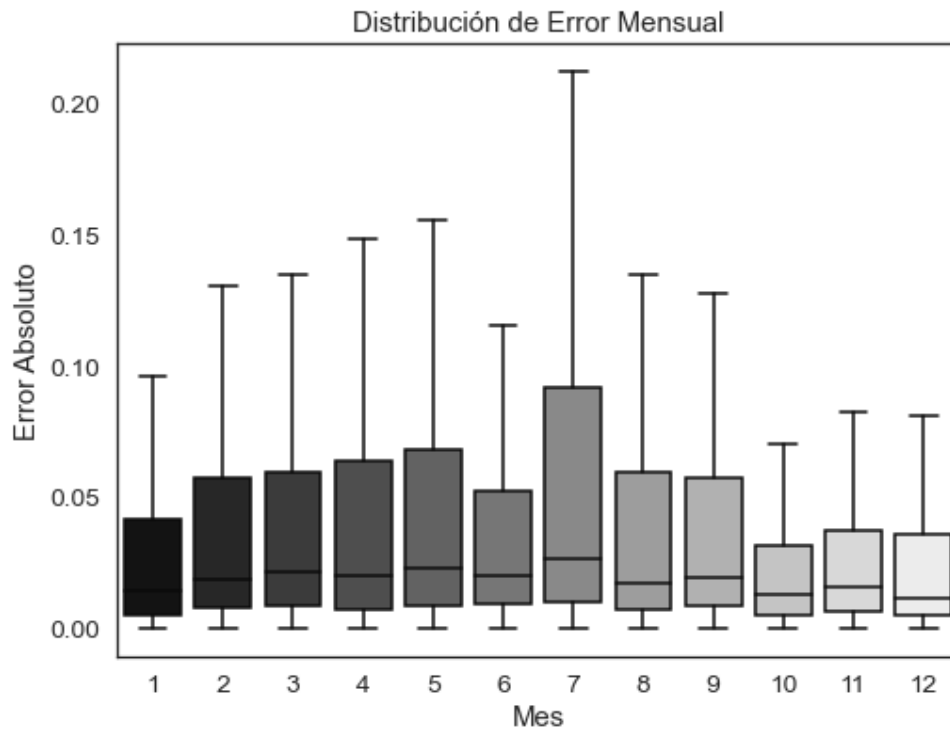


Figura 4.10: Gráfica de dispersión relacional



(a) Con valores atípicos



(b) Sin valores atípicos

Figura 4.11: Diagramas de cajas y bigotes del error absoluto mensual



## 4. 4. Evaluación de resultados

Los resultados globales de las métricas de desempeño obtenidas por el modelo propuesto se muestran en la tabla 4.5 en la columna "Valor Obtenido", mientras que los valores de referencia, es decir, resultados reportados en trabajos similares con los cuales se comparan los resultados del modelo propuesto, se muestran en la columna "Valor de Referencia".

Métrica	Valor Obtenido	Valor de Referencia
Error absoluto medio (MAE)	0.0486 °C	0.18 °C [38]
Error cuadrático medio (MSE)	0.0095 °C <sup>2</sup>	0.2659 [31]
Raíz cuadrada del error cuadrático medio (RMSE)	0.0976 °C	0.23 °C [38]
Coefficiente de determinación (R <sup>2</sup> )	0.9985	0.971 [31]

Tabla 4.5: Resultados de las métricas

# Capítulo 5

## Conclusiones

Con base en los resultados de las métricas de desempeño en la evaluación de los modelos de redes neuronales artificiales, es posible confirmar la hipótesis planteada en este trabajo de tesis. Toda vez que los valores obtenidos mejoran los resultados reportados en otros trabajos donde también se emplean redes neuronales artificiales para realizar predicción, como los artículos [38], [31], por citar algunos.

En el modelo implementado se tiene la certeza de haber seleccionado la mejor combinación de número de capas ocultas y número de neuronas por capa oculta, ya que ambos parámetros fueron determinados por experimentación, seleccionando la combinación que produjo el menor valor de la función de pérdida con los datos de validación.

En relación a los datos *in situ* que se obtuvieron, fueron divididos en la etapa de entrenamiento considerando que la ubicación geográfica de las mediciones quedara representada en cada uno de los tres subconjuntos creados.

Como trabajos futuros, se propone adecuar el modelo propuesto para la predicción de los perfiles completos, es decir, en toda la columna de agua. Además, implementar modelos para otros parámetros hidrográficos como el oxígeno. Posteriormente, implementar los modelos para pronosticar condiciones futuras a mediano y largo plazo utilizando como primera opción las redes neuronales recurrentes, sin restringir el poder utilizar alguna otra técnica de aprendizaje automático que pueda contribuir a la mejora de los resultados. Por último, se plantea el poder utilizar datos de satélite, los cuales tiene la ventaja de contar con una mejor resolución temporal y espacial, para reproducir los perfiles y utilizar las mediciones *in situ* para validar los resultados.

# Apéndice A

## Mediciones mensuales

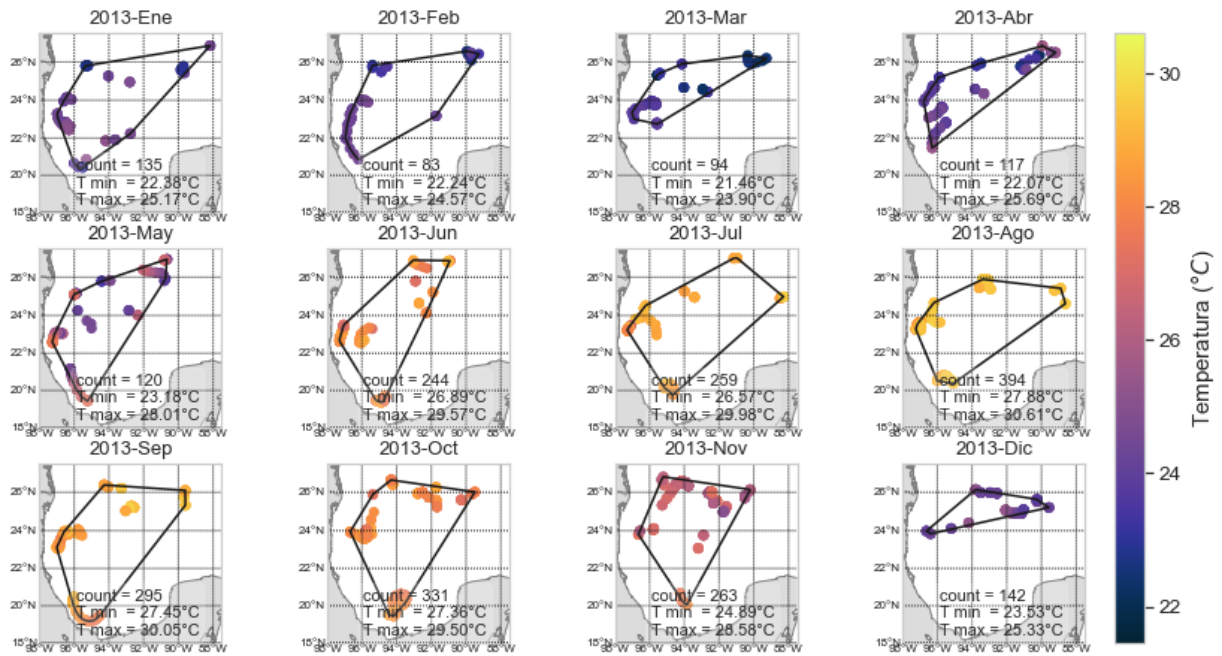
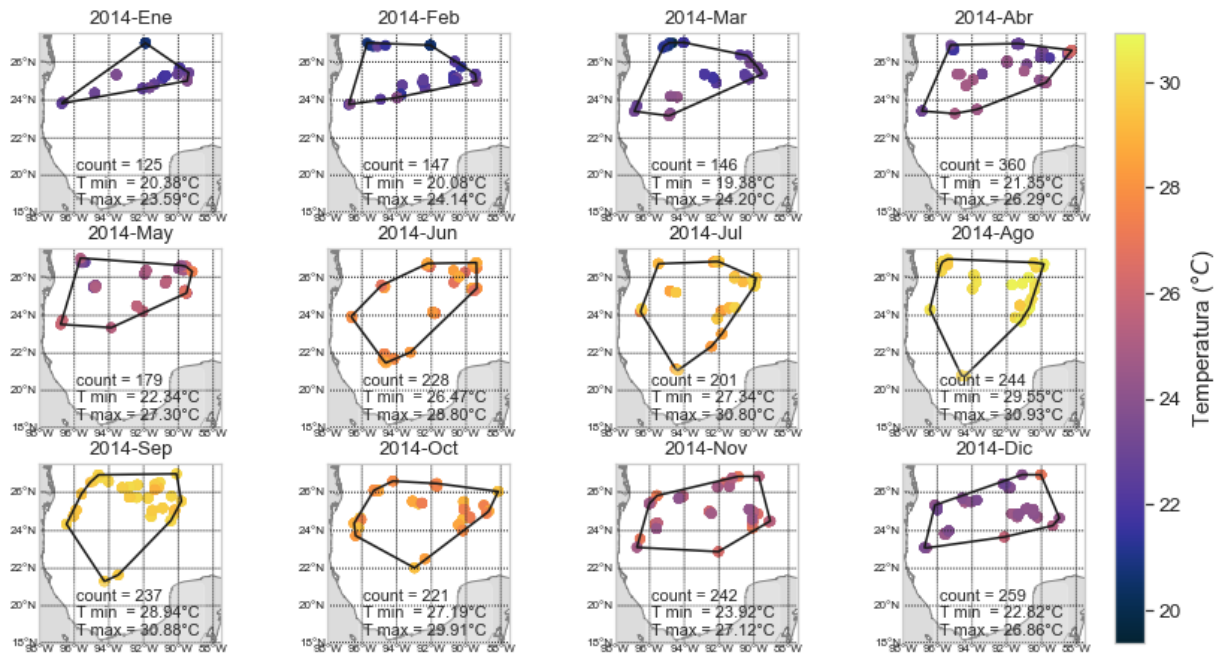
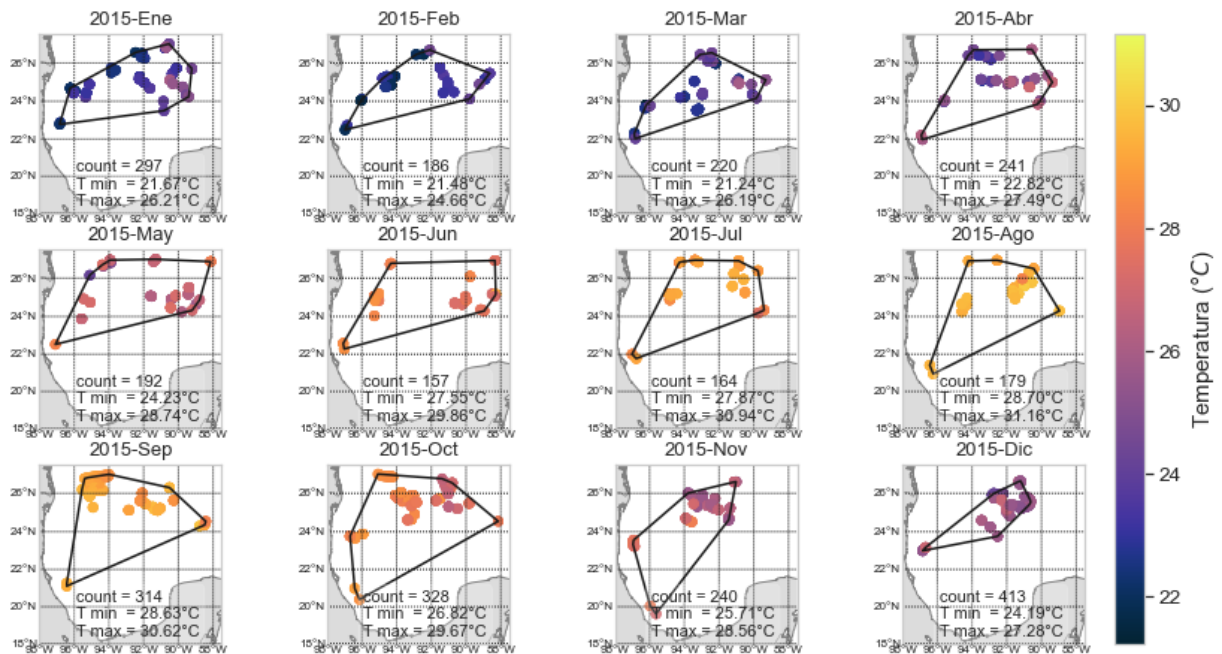


Figura A.1: Mediciones del año 2013 con profundidad de 0 a 10 metros. “count” indica el número de mediciones disponibles, “T min” el valor mínimo, “T max” el valor máximo

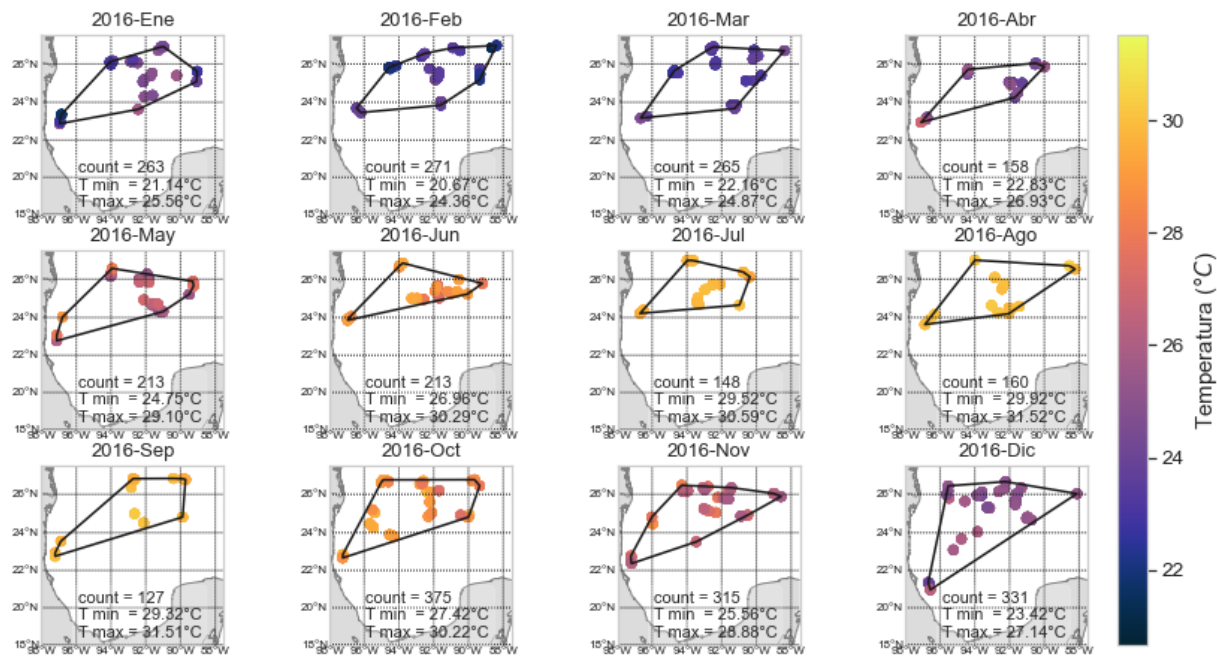


(a) 2014

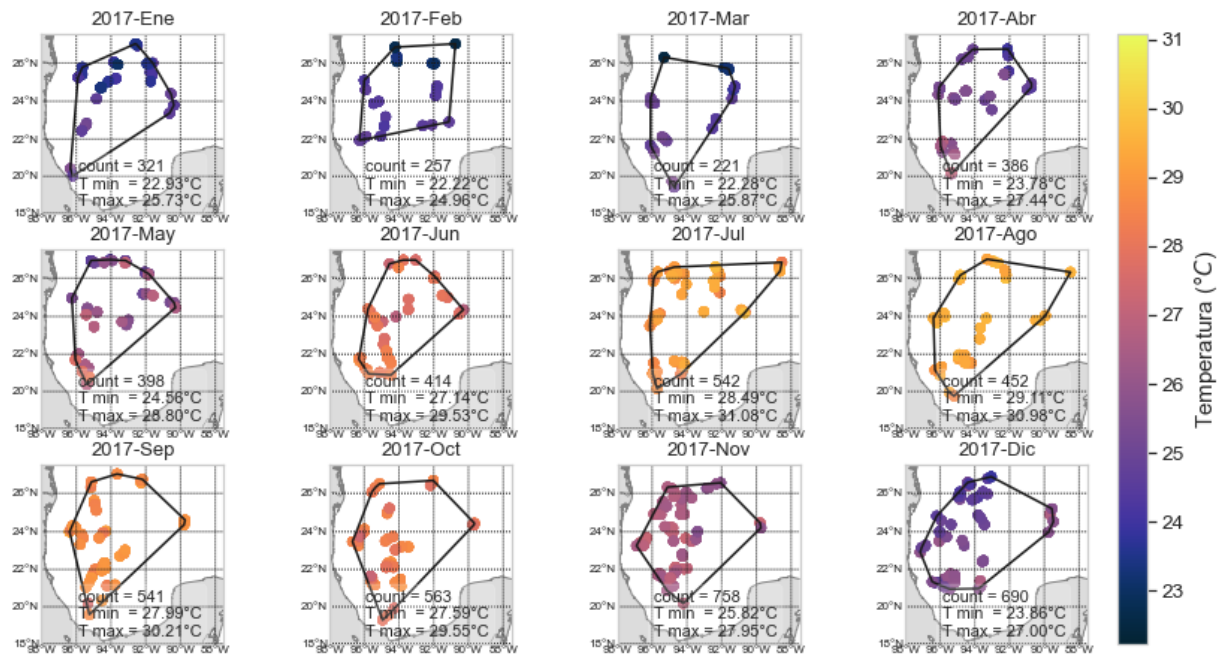


(b) 2015

Figura A.2: Mediciones de los años 2014 y 2015 con profundidad de 0 a 10 metros. “count” indica el número de mediciones disponibles, “T min” el valor mínimo, “T max” el valor máximo

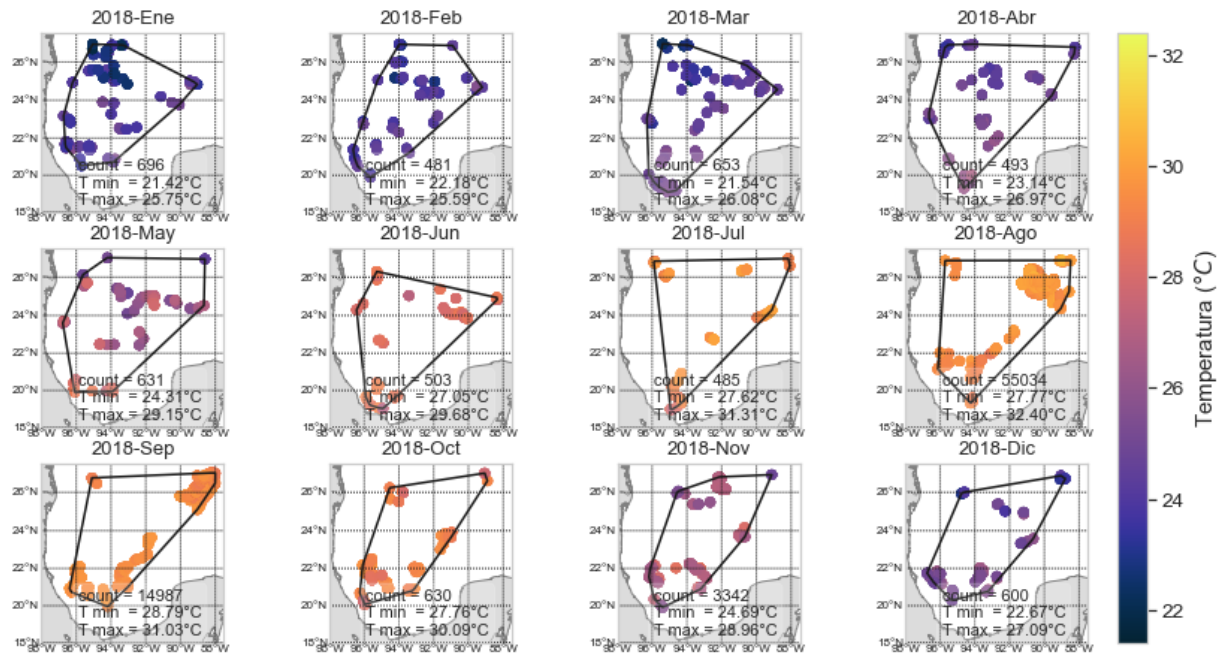


(a) 2016

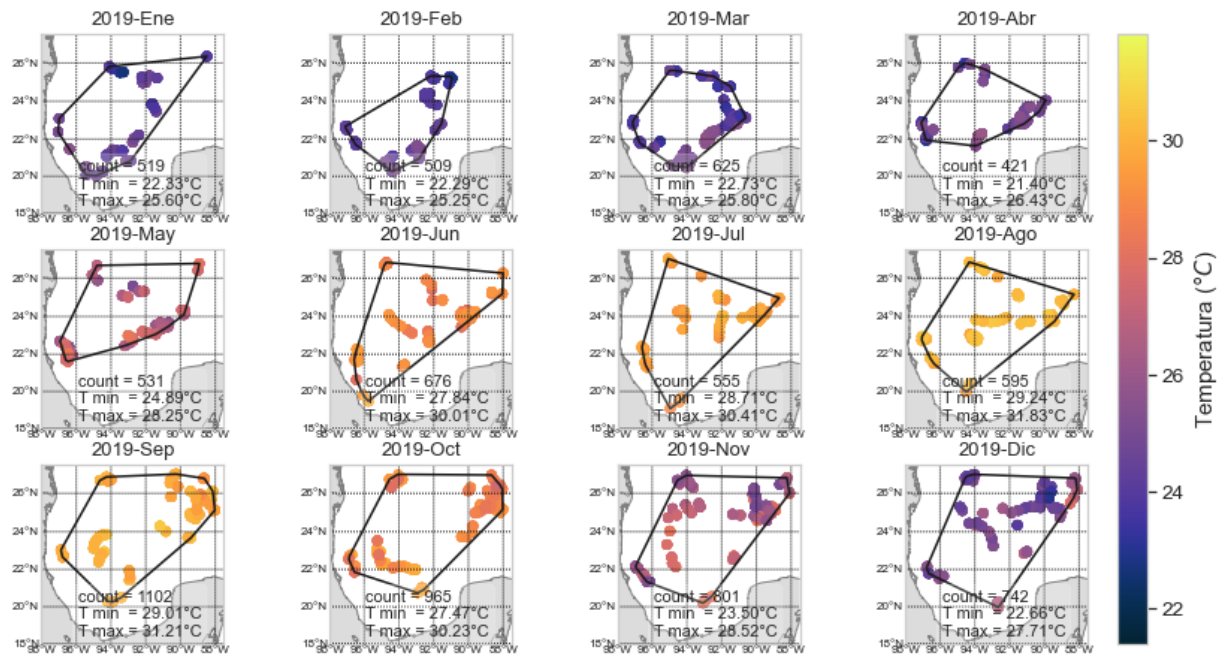


(b) 2017

Figura A.3: Mediciones de los años 2016 y 2017 con profundidad de 0 a 10 metros. “count” indica el número de mediciones disponibles, “T min” el valor mínimo, “T max” el valor máximo

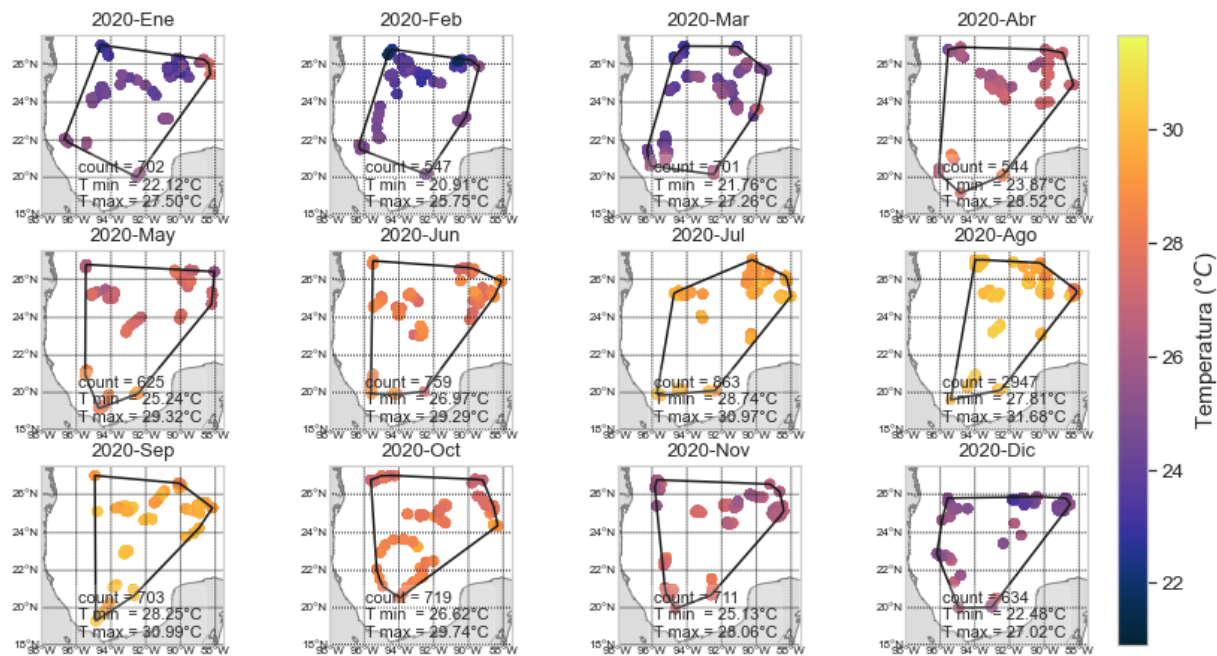


(a) 2018

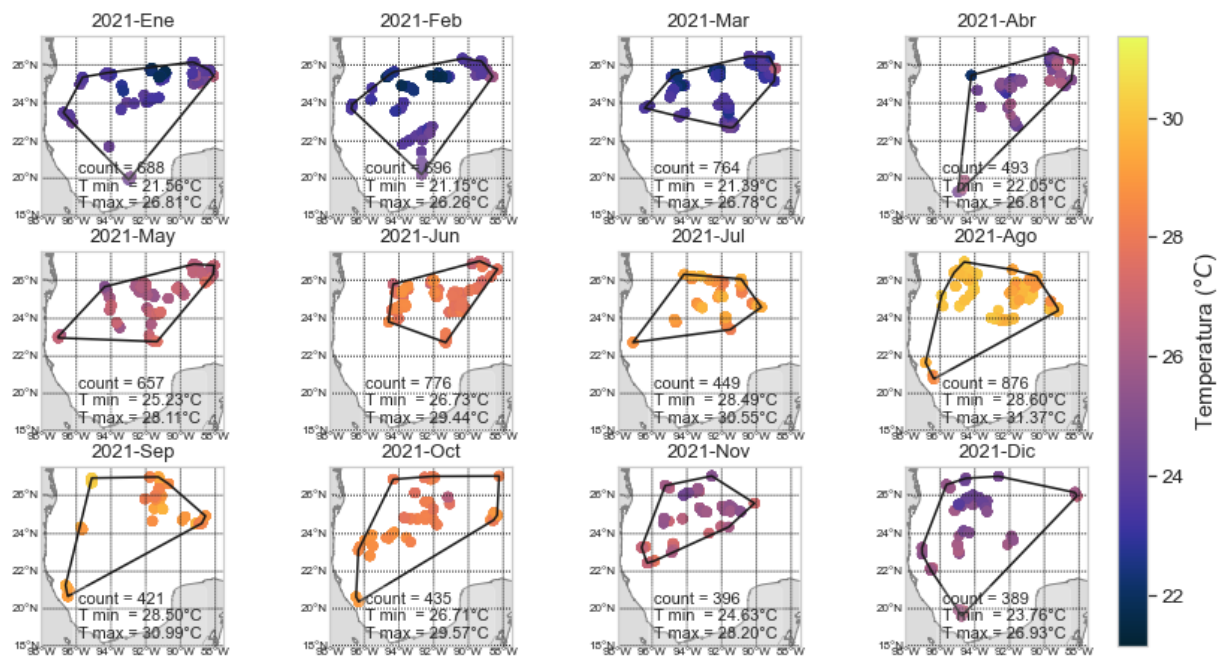


(b) 2019

Figura A.4: Mediciones de los años 2018 y 2019 con profundidad de 0 a 10 metros. “count” indica el número de mediciones disponibles, “T min” el valor mínimo, “T max” el valor máximo



(a) 2020



(b) 2021

Figura A.5: Mediciones de los años 2020 y 2021 con profundidad de 0 a 10 metros. “count” indica el número de mediciones disponibles, “T min” el valor mínimo, “T max” el valor máximo

# Apéndice B

## Predicciones espaciales mensuales

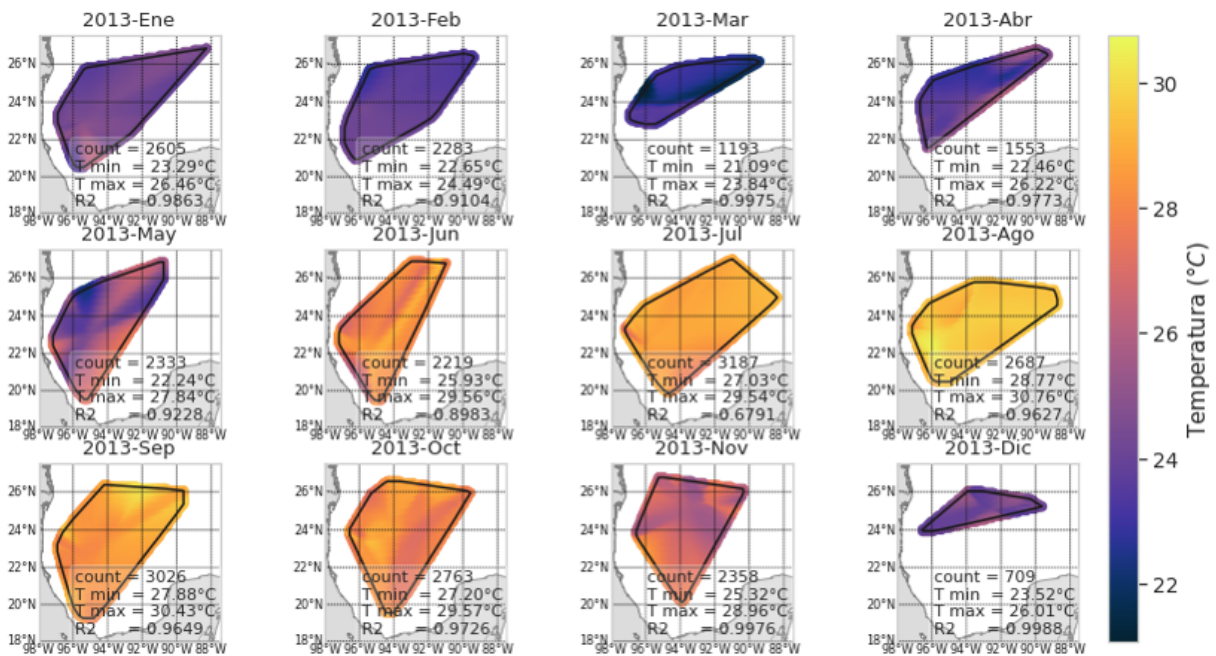
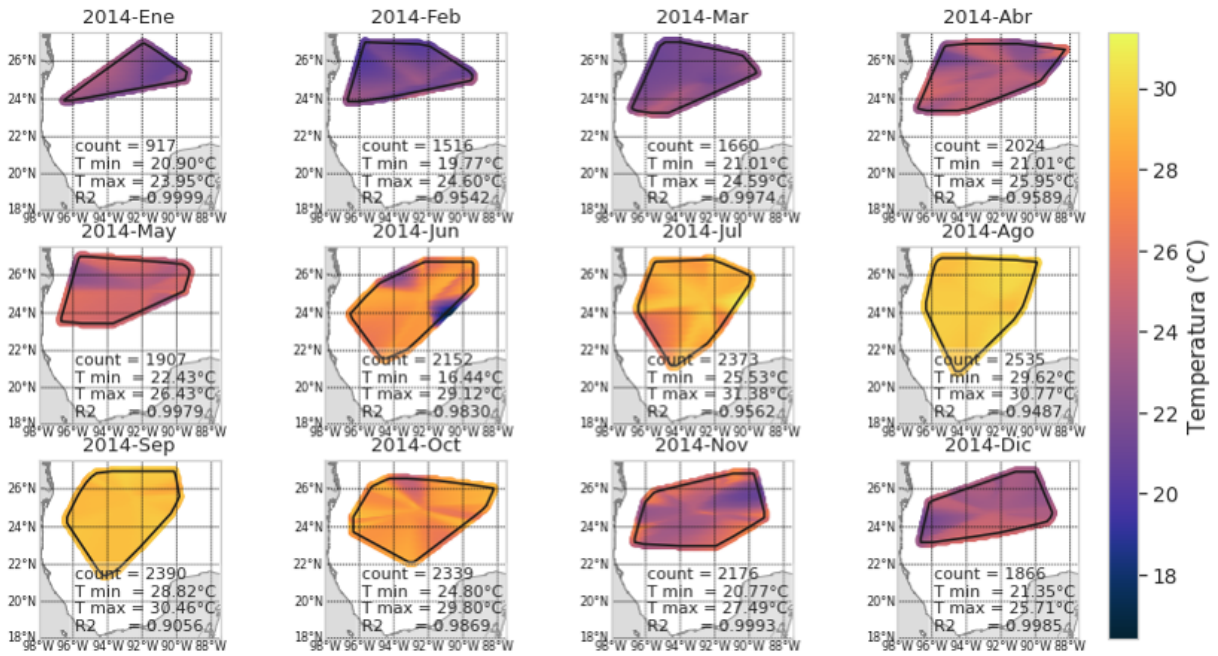
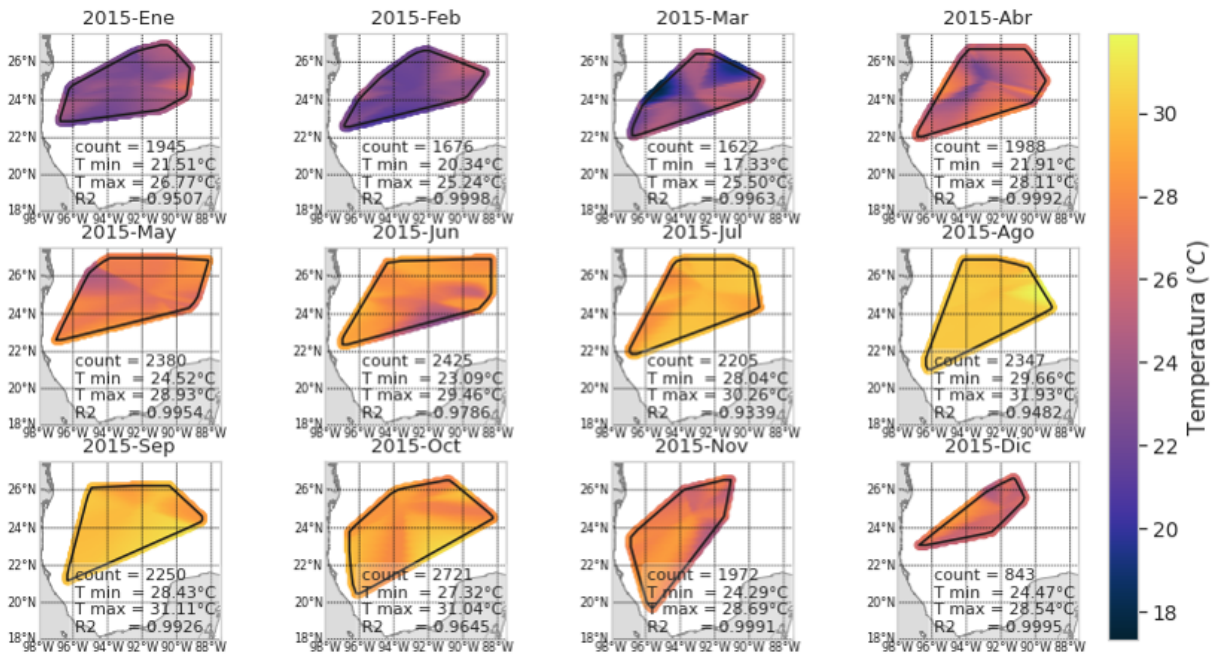


Figura B.1: Predicciones del año 2013. “count” indica el número de mediciones disponibles, “T min” el valor mínimo, “T max” el valor máximo



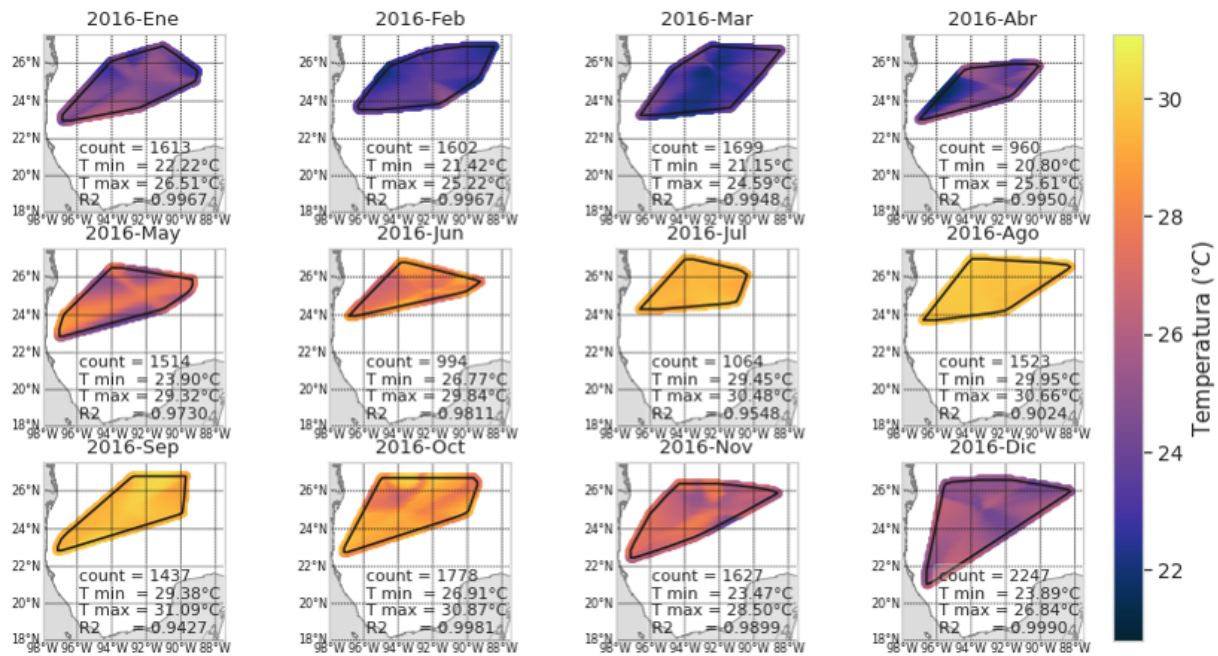


(a) 2014

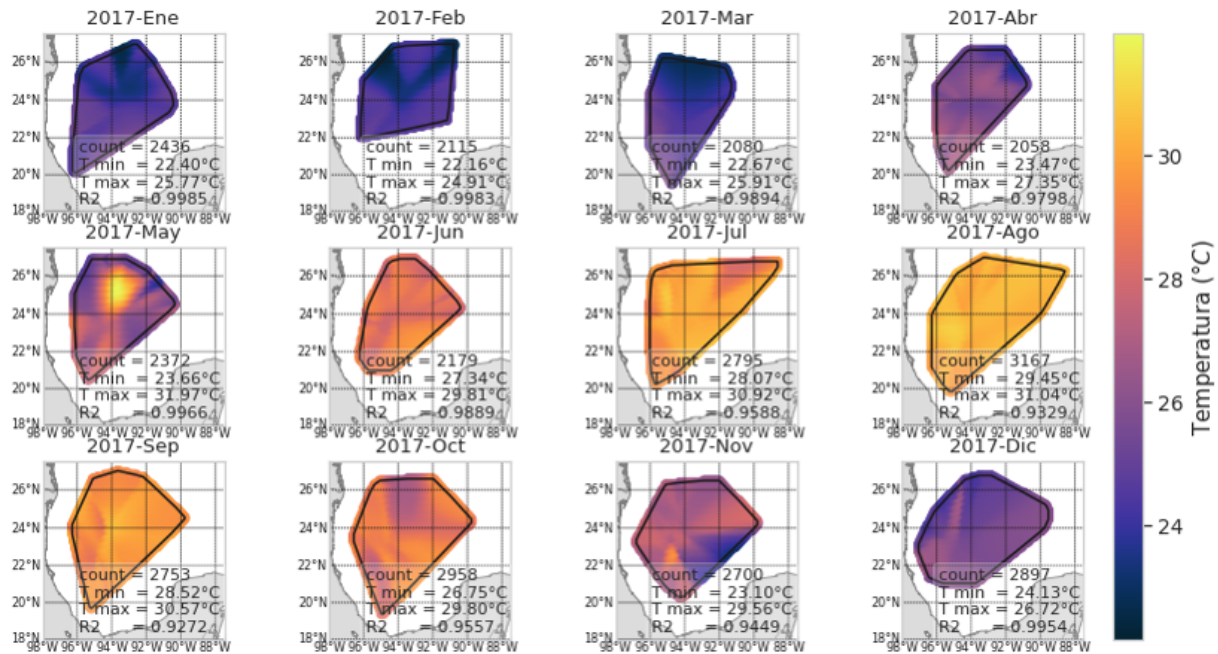


(b) 2015

Figura B.2: Predicciones de los años 2014 y 2015. “count” indica el número de mediciones disponibles, “T min” el valor mínimo, “T max” el valor máximo

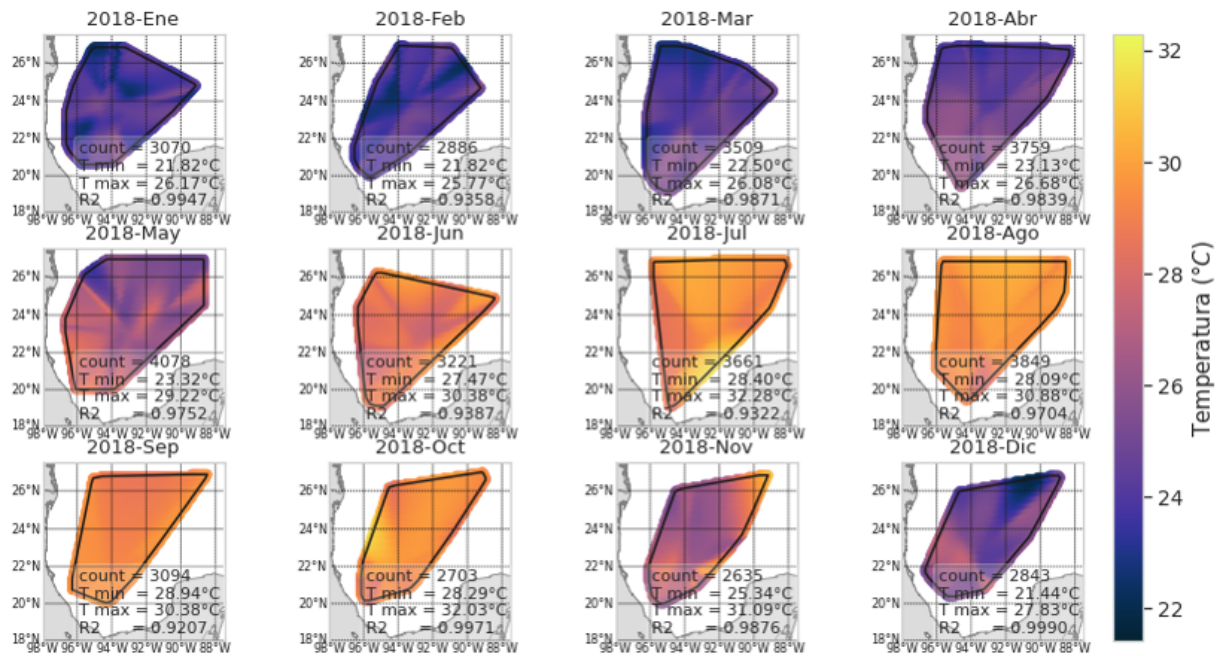


(a) 2016

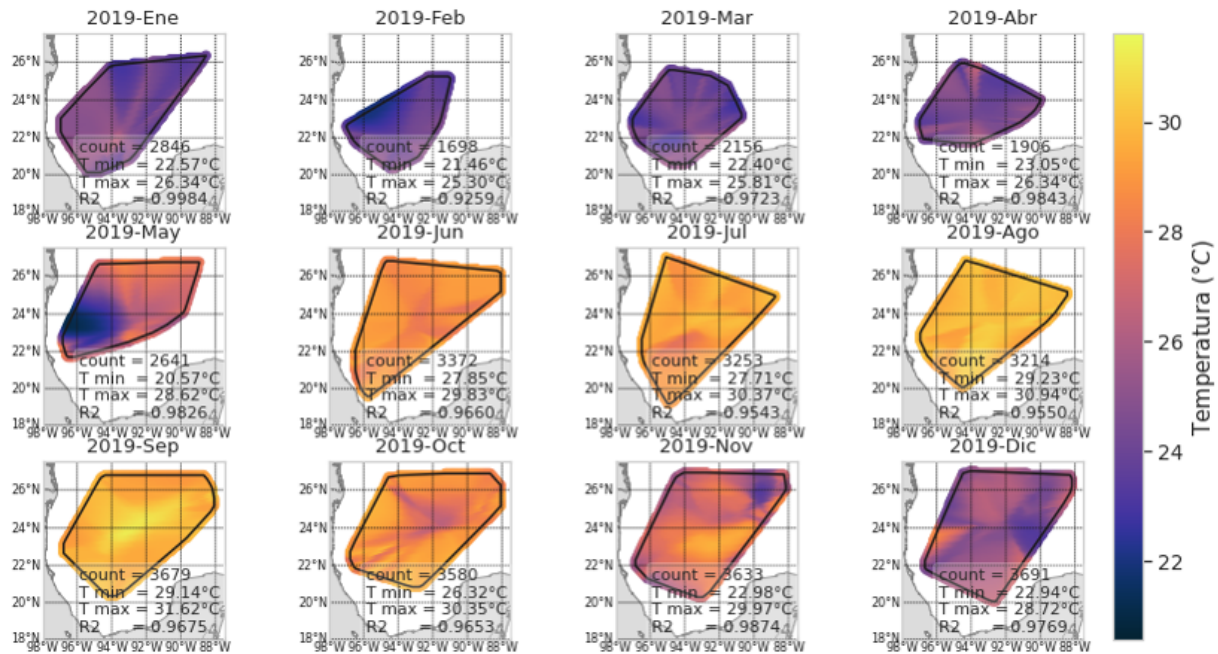


(b) 2017

Figura B.3: Predicciones de los años 2016 y 2017. “count” indica el número de mediciones disponibles, “T min” el valor mínimo, “T max” el valor máximo

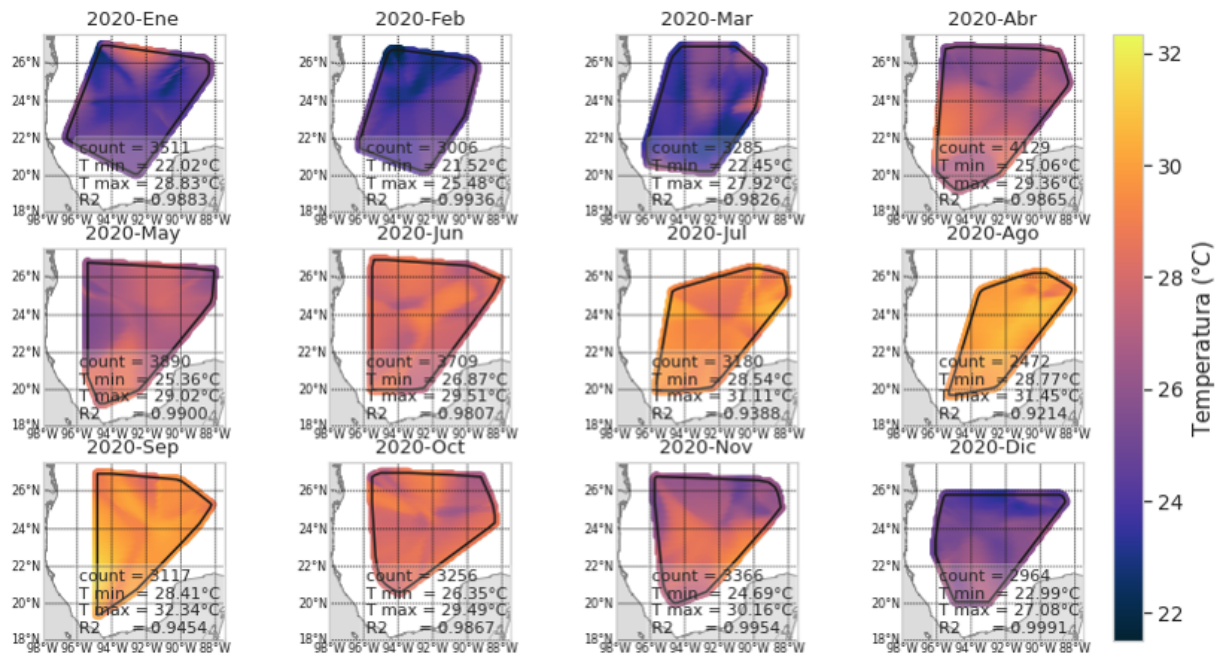


(a) 2018

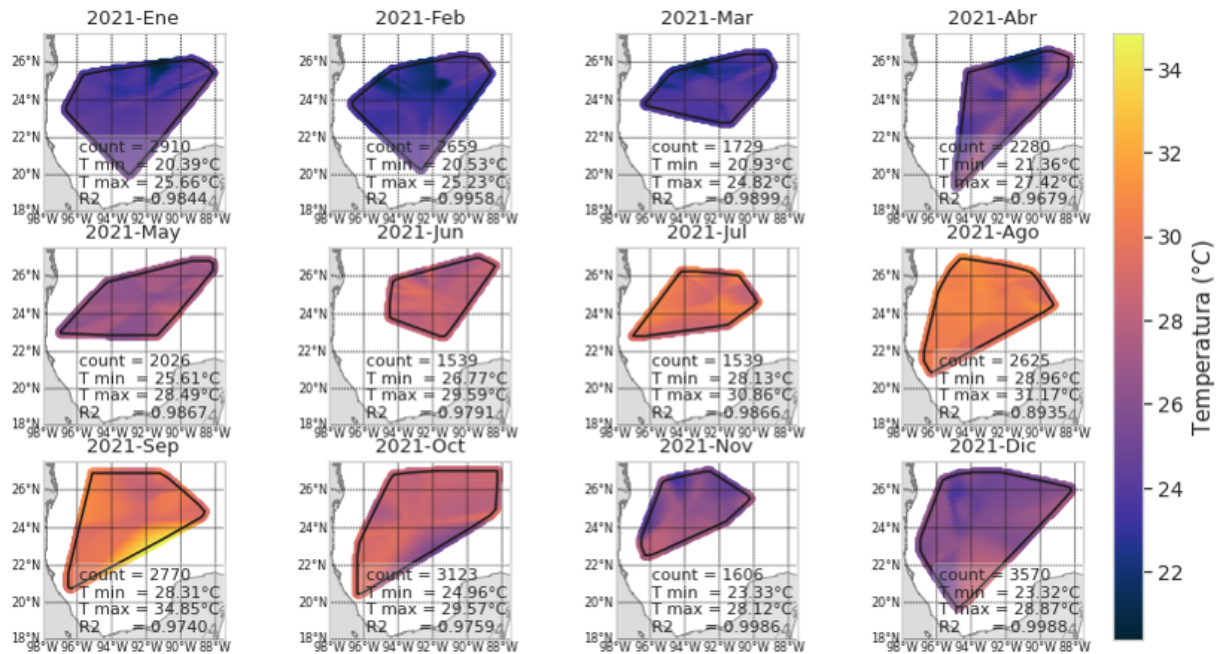


(b) 2019

Figura B.4: Predicciones de los años 2018 y 2019. “count” indica el número de mediciones disponibles, “T min” el valor mínimo, “T max” el valor máximo



(a) 2020



(b) 2021

Figura B.5: Predicciones de los años 2020 y 2021. “count” indica el número de mediciones disponibles, “T min” el valor mínimo, “T max” el valor máximo

# Bibliografía

- [1] GOOS Observations Coordination Group. A 5-year Strategic Plan for OceanOPS 2021-2025. Technical report, 2021.
- [2] World Meteorological Organization. WMO - Oceans, 2021.
- [3] GOOS. The Global Ocean Observing System. Web page., 2021.
- [4] Tim P.; Boyer, Olga K.; Baranova, Coleman; Carla, Hernan E.; Garcia, Alexandra; Grodsky, Ricardo A.; Locarnini, Alexey V.; Mishonov., Christopher R.; Paver., James R.; Reagan., Dan; Seidov., Igor V.; Smolyar., Katharine W.; Weathers., and Melissa M.; Zweng. *NOAA Atlas NESDIS 87*. 2018.
- [5] H. E.. Garcia, T. P. Boyer., R. A. Locarnini., O. K. Baranova., and M. M. Zweng. *World Ocean Database 2018. User's Manual*. 2018.
- [6] Argo. Argo Web Page, 2021.
- [7] Esther Portela, Miguel Tenreiro, Enric Pallàs-Sanz, Thomas Meunier, Angel Ruiz-Angulo, Rossmery Sosa-Gutiérrez, and Simo Cusi. Hydrography of the Central and Western Gulf of Mexico. *Journal of Geophysical Research: Oceans*, 123, 2018.
- [8] E Romero, L Tenorio-Fernandez, I Castro, and M Castro. Filtering method based on cluster analysis to avoid salinity drifts and recover Argo data in less time. *Ocean Science*, 17(5):1273–1284, 2021.
- [9] Ganda Yoga Swara Anisya. Implementation Of Haversine Formula And Best First Search Method In Searching Of Tsunami Evacuation Route. *IOP Conference Series: Earth and Environmental Science*, 2017.

- [10] P Dauni, M D Firdaus, R Asfariani, M I N Saputra, A A Hidayat, and W B Zulfikar. Implementation of Haversine formula for school location tracking. *Journal of Physics: Conference Series*, 1402(7):77028, 2019.
- [11] Rakhmanov Ochilbek. A new approach (extra vertex) and generalization of Shoelace Algorithm usage in convex polygon (Point-in-Polygon). 2018.
- [12] Satyan L.; Devadoss and Joseph; O'Rourke. *Discrete and Computational GEOMETRY*. 2011.
- [13] Sahoo K. et al. Exploratory data analysis using python. *International Journal of Innovative Technology and Exploring Engineering*, 8(12):4727–4735, 2019.
- [14] Argo. Argo float data and metadata from Global Data Assembly Centre (Argo GDAC)., 2022.
- [15] Douglas C. Montgomery, Cheryl L. Jennings, and Murat Kulahci. *Introduction to Time Series Analysis and Forecasting*. 2007.
- [16] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, and Grata M. Ljung. *Time Series Analysis Forecasting and Control*. 2016.
- [17] Aileen Nielsen. *Practical Time Series Analysis Prediction with Statistics and Machine Learning*. 2019.
- [18] Aurélien Gerón. *Hands-on Machine Learning with Scikit-Learn, Keras and TensorFlow*. 2019.
- [19] Jason Brownlee. *Clever Algorithms. Nature-Inspired Programming Recipes*. 2012.
- [20] Jason Bell. *Machine Learning Hands-On For Developers and Technical Professionals*. Wiley, 2015.
- [21] Filippo Maria. Bianchi, Maiorino Enrico., Michael Kampffmeyer, Antonello. Rizzi, and Robert Jenssen. *Recurrent Neural Networks for Short-Term Load Forecasting. An Overview and Comparative Analysis*. 2017.

- [22] Spyros Makridakis, Evangelos Spiliotis, and Vassilios Assimakopoulos. The M4 Competition: Results, findings, conclusion and way forward. *International Journal of Forecasting*, 34(4):802–808, 2018.
- [23] Bryan; Lim and Stefan; Zohren. Time Series Forecasting With Deep Learning: A Survey. 2020.
- [24] José F. Torres, Dalil Hadjout, Abderrazak Sebaa, Francisco Martínez-Álvarez, and Alicia Troncoso. Deep Learning for Time Series Forecasting: A Survey. *Big Data*, 9(1):3–21, 2021.
- [25] Ahmed Ali, Ahmed Fathalla, Ahmad Salah, Mahmoud Bekhit, and Esraa Eldesouky. Marine Data Prediction: An Evaluation of Machine Learning, Deep Learning, and Statistical Predictive Models. *Computational Intelligence and Neuroscience*, 2021:8551167, 2021.
- [26] Kalpesh Patil and M C Deo. Basin-Scale Prediction of Sea Surface Temperature with Artificial Neural Networks. *Journal of Atmospheric and Oceanic Technology*, 35(7):1441–1455, 2018.
- [27] Aparna Sairesh Gandhi. Prediction of daily sea surface temperature using artificial neural networks. *ResearchGate*, 2018.
- [28] Gábor Petneházi. Recurrent Neural Networks for Time Series Forecasting. 2019.
- [29] Su Hua, Zhang Tianyi, Lin Mengjing, Lu Wenfang, and Yan Xiao-Hai. Predicting subsurface thermohaline structure from remote sensing data based on long short-term memory neural networks. *ELSEVIER*, 2021.
- [30] Steefan Contractor and Moninya Roughan. Efficacy of Feedforward and LSTM Neural Networks at Predicting and Gap Filling Coastal Ocean Timeseries: Oxygen, Nutrients, and Temperature. *Frontiers in Marine Science*, 8, 2021.
- [31] Mingxu Han, Yuan Feng, Xueli Zhao, Chunjian Sun, Feng Hong, and Chao Liu. A Convolutional Neural Network Using Surface Data to Predict Subsurface Temperatures in the Pacific Ocean. *IEEE Access*, 7:172816–172829, 2019.
- [32] Lingsheng Meng, Chi Yan, Wei Zhuang, Weiwei Zhang, Xupu Geng, and Xiao-Hai Yan. Reconstructing High-Resolution Ocean Subsurface and Interior Temperature and Salinity

- Anomalies From Satellite Observations. *IEEE Transactions on Geoscience and Remote Sensing*, 60:1–14, 2021.
- [33] Zao; Zhang and Yuan; Dong. Temperature Forecasting via Convolutional Recurrent Neural Networks Based on Time-Series Data. *Hindawi*, 2020.
- [34] Zhiyuan Wu, Changbo Jiang, Mack Conde, Bin Deng, and Jie Chen. Hybrid improved empirical mode decomposition and BP neural network model for the prediction of sea surface temperature. *Ocean Science*, 15(2):349–360, 2019.
- [35] Argo. Argo GDAC - Snapshot, 2021.
- [36] World Ocean Database Website, 2022.
- [37] Autonomio Talos. Computer Software, 2020.
- [38] Kalpesh Patil and Makaranad Chintamani Deo. Prediction of daily sea surface temperature using efficient neural networks. *Ocean Dynamics*, 67(3):357–368, 2017.