

SEP

TNM

INSTITUTO TECNOLÓGICO DE CULIACÁN



PRONÓSTICO DE LA VELOCIDAD DEL VIENTO MEDIANTE  
TÉCNICAS DE INTELIGENCIA ARTIFICIAL

TESIS

PRESENTADA ANTE EL DEPARTAMENTO ACADÉMICO DE ESTUDIOS DE POSGRADO  
DEL INSTITUTO TECNOLÓGICO DE CULIACÁN EN CUMPLIMIENTO PARCIAL DE LOS  
REQUISITOS PARA OBTENER EL GRADO DE

MAESTRO EN CIENCIAS DE LA COMPUTACIÓN

POR:

JUAN RAMÓN VALENZUELA BARRAZA  
INGENIERO MECATRÓNICO

DIRECTOR DE TESIS:  
HÉCTOR RODRÍGUEZ RANGEL

CULIACÁN, SINALOA

9 de Agosto del 2019



“2019, Año del Caudillo del Sur, Emiliano Zapata”

Culiacán, Sin., 7 de Agosto del 2019

**DIVISIÓN DE ESTUDIOS DE POSGRADO E INVESTIGACIÓN**

OFICIO: DEPI:335/VIII/2019

ASUNTO: **Autorización Impresión**

**ING. JUAN RAMÓN VALENZUELA BARRAZA**  
**ESTUDIANTE DE LA MAESTRÍA EN CIENCIAS DE LA COMPUTACIÓN**  
**PRESENTE.**

Por medio de la presente y en virtud de que ha completado los requisitos para el examen de grado de la **Maestría en Ciencias de la Computación**, se concede autorización para la impresión de la tesis titulada: **“PRONÓSTICO DE LA VELOCIDAD DEL VIENTO MEDIANTE TÉCNICAS DE INTELIGENCIA ARTIFICIAL”** bajo la dirección del(a) **Dr. Héctor Rodríguez Rangel**

Sin otro particular reciba un cordial saludo.

**ATENTAMENTE**  
*Excelencia en Educación Tecnológica®*

**M.C. MARÍA ARACELY MARTÍNEZ AMAYA**  
**JEFE(A) DE LA DIVISIÓN DE ESTUDIOS DE**  
**POSGRADO E INVESTIGACIÓN**

SEP TecNM  
Instituto Tecnológico  
de Culiacán  
División de Estudios  
de Posgrado e Investigación

C.c.p. archivo

MAMA/lucy \*





# “PRONÓSTICO DE LA VELOCIDAD DEL VIENTO MEDIANTE TÉCNICAS DE INTELIGENCIA ARTIFICIAL”

Tesis presentada por:

**ING. JUAN RAMÓN VALENZUELA BARRAZA**

Aprobada en contenido y estilo por:

Dr. Héctor Rodríguez Rangel  
Director de Tesis

Dra. María Lucía Barrón Estrada  
Secretario

Dr. Ramón Zatarain Cabada  
Vocal -1

Dr. Víctor Alejandro González Huitrón  
Vocal -2

M.C. María Aracely Martínez Amaya  
Jefe(a) de la División de Estudios de Posgrado e Investigación



# Dedicatoria

Dedico este trabajo a mis padres que con apoyo incondicional, amor y confianza permitieron que logre alcanzar mis metas. Por siempre estar a mi lado y por llenar mi vida con sus valiosos consejos. Gracias Juan Manuel Valenzuela Bátiz y Maria de los Angeles Barraza Sandoval.

# Agradecimientos

## **A dios**

Agradezco a Dios por guiarme en mi camino y por permitirme concluir con mi objetivo.

## **A mis padres**

Por todos sus consejos y apoyo incondicional.

## **A mis hermanos**

Con quienes he compartido hermosos momentos en mi vida y siempre me han apoyado.

## **A CONACyT**

Por hacer posible realizar este trabajo.

## **A mi director de tesis y asesores**

Quienes con sus conocimientos y su gran trayectoria, han logrado en mí culminar mis estudios con éxito.

## **A la institución**

Al Instituto Tecnológico de Culiacán y al departamento de Posgrado, por la formación que me han dado.

## **A mis compañeros**

Por los consejos y buenos momentos que compartimos durante nuestros estudios.

# Declaración de Autenticidad

Por la presente declaro que, salvo cuando se haga referencia específica al trabajo de otras personas, el contenido de esta tesis es original y no se ha presentado total o parcialmente para su consideración para cualquier otro título o grado en esta o cualquier otra Universidad. Esta tesis es resultado de mi propio trabajo y no incluye nada que sea resultado de algún trabajo realizado en colaboración, salvo que se indique específicamente en el texto.

Juan Ramón Valenzuela Barraza. Culiacán, Sinaloa, México, 2019.

# Resumen

La energía eléctrica convencional consumida por la población es una necesidad en las sociedades moderna, esto debido a que las actividades realizadas por la humanidad cada ves son más dependientes de procesos automatizados. Esto causa la necesidad de producir de energía eléctrica suficiente, la cual puede ser generada de múltiples maneras. Entre las distintas formas de producción de electricidad más de dos tercios de esta energía es producida por medios fósiles, los cuales se caracterizan por generar gases contaminantes que afectan el medio ambiente y la salud de la población.

Una de las alternativas para producir energía eléctrica que posee un menor impacto negativo hacia el ambiente es la energía eólica. Esta energía se genera a partir de la rotación mecánica generada por el viento que golpea las aspas de una turbina. La energía eólica posee la ventaja de no generar gases contaminantes; además, esta ha ganado popularidad en años recientes, lo cual ha impulsado la evolución y avance tecnológico utilizada en aerogeneradores. Sin embargo, el viento que es utilizado como recurso primario para la producción de energía eólica tiene un comportamiento caótico que puede variar constantemente, lo cual dificulta la producción en las granjas eólicas. Es por esto que es necesario el uso de modelos pronosticadores que faciliten la producción de energía eólica al proveer una proyección del comportamiento que tendrá el viento.

En esta Tesis se trabajo con el pronóstico de la velocidad del viento a corto plazo, específicamente 24 horas. Se diseño una metodología que combina modelos de Inteligencia Artificial (IA) con estrategias de pronósticos a múltiples pasos. Los modelos de IA utilizados para este trabajo fueron: vecinos más cercanos ( $k$ -NN), Perceptron Multicapas (MLP), Redes de gran Memoria a Corto Plazo (LSTM) y Redes Neuronales Convolucionales (CNN). Las estrategias utilizadas son: recursiva, directa, MIMO (*Multiple Inputs Multiple Outputs*), DIRMO (combinación de directa y MIMO) y DirREC (combinación de directa y recursiva).

Para el entrenamiento y validación de los modelos de pronóstico se aplicaron técnicas de preprocesamiento a los datos para reducir el ruido y datos atípicos. Finalmente, se seleccionaron los modelos con mejor aptitud para optimizar sus parámetros por medio de algoritmos genéticos inicializados por búsqueda aleatoria.

# Palabras Clave

- Aprendizaje máquina
- Aprendizaje profundo
- Estrategias de pronóstico
- Horizonte de pronóstico
- Inteligencia artificial
- Optimización de hiperparámetros
- Velocidad del viento



# Índice general

---

<b>Índice de figuras</b>	<b>X</b>
<b>Índice de tablas</b>	<b>XIII</b>
<b>1. Introducción</b>	<b>1</b>
1.1. Definición del problema . . . . .	3
1.2. Hipótesis . . . . .	4
1.3. Objetivo . . . . .	4
1.4. Objetivos particulares . . . . .	5
1.5. Justificación . . . . .	5
1.6. Estructura de la tesis . . . . .	7
<b>2. Marco teórico</b>	<b>8</b>
2.1. Energía eólica . . . . .	8
2.2. Pronóstico . . . . .	9
2.2.1. Serie de tiempo . . . . .	10
2.2.2. Pronóstico a un paso en el tiempo . . . . .	11
2.2.3. Pronóstico a múltiples pasos en el tiempo . . . . .	12
2.2.4. Estrategias para pronósticos de múltiples pasos en el tiempo . . . . .	12
2.2.4.1. Estrategia Recursiva . . . . .	14
2.2.4.2. Estrategia Directa . . . . .	16
2.2.4.3. Estrategia MIMO . . . . .	17
2.2.4.4. Estrategia DirREC . . . . .	18
2.2.4.5. Estrategia DIRMO . . . . .	20
2.3. Inteligencia Artificial . . . . .	21
2.3.1. Aprendizaje Máquina . . . . .	22
2.3.2. k Vecinos más cercanos . . . . .	24
2.3.3. Aprendizaje Profundo . . . . .	24
2.3.3.1. Perceptrón Multicapas . . . . .	26
2.3.3.2. Redes de gran memoria a corto plazo . . . . .	27
2.3.3.3. Redes Neuronales Convolucionadas . . . . .	29
2.4. Optimización de hiperparámetros . . . . .	31
2.4.1. Algoritmos Genéticos . . . . .	32
2.4.2. Optimización de hiperparámetros por caja negra . . . . .	35
2.5. Métricas estadísticas . . . . .	36

<b>3. Estado del Arte</b>	<b>38</b>
3.1. Métodos de pronóstico . . . . .	38
3.2. Métodos de optimización de hiperparámetros . . . . .	43
<b>4. Metodología Propuesta</b>	<b>44</b>
4.1. Preprocesamiento de datos . . . . .	46
4.1.1. Limpieza de datos . . . . .	46
4.1.1.1. Normalización de datos . . . . .	47
4.1.1.2. Media Móvil . . . . .	48
4.1.2. Transformación de datos . . . . .	49
4.1.2.1. Transformación de serie de tiempo a secuencias . . . . .	49
4.1.2.2. Transformación de de serie de tiempo a imagen . . . . .	51
4.2. Implementación de Estrategias de predicción múltiples pasos en el tiempo . .	52
4.2.1. Estrategia Recursiva . . . . .	53
4.2.2. Estrategia Directa . . . . .	54
4.2.3. Estrategia MIMO . . . . .	55
4.2.4. Estrategia DIRMO . . . . .	56
4.2.5. Estrategia DirREC . . . . .	57
4.3. Modelos de Inteligencia Artificial . . . . .	59
4.3.1. Hiperparámetros de implementación del modelo MLP . . . . .	61
4.3.2. Hiperparámetros de implementación del modelo LSTM . . . . .	62
4.3.3. Hiperparámetros de implementación del modelo LSTM-MLP . . . . .	63
4.3.4. Hiperparámetros de implementación del modelo CNN . . . . .	64
4.3.5. Hiperparámetros de implementación del modelo CNN-LSTM . . . . .	65
4.3.6. Hiperparámetros de implementación del modelo CNN-LSTM-ANN .	67
4.4. Optimización de hiperparámetros . . . . .	68
4.5. Tecnologías utilizadas . . . . .	70
4.5.1. Python . . . . .	70
4.5.2. Tensorflow . . . . .	70
4.5.3. Keras . . . . .	70
4.5.4. Docker . . . . .	71
<b>5. Análisis de Resultados</b>	<b>72</b>
5.1. Resultados de Aristeo Mercado . . . . .	73
5.2. Resultados de Cointzio . . . . .	75
5.3. Resultados Corrales . . . . .	77
5.4. Resultados La Piedad . . . . .	79
5.5. Resultados Melchor Campo . . . . .	80
<b>6. Conclusiones</b>	<b>83</b>
6.1. Conclusiones . . . . .	83
6.2. Aportaciones . . . . .	84
6.3. Trabajo Futuro . . . . .	85
<b>Bibliografía</b>	<b>86</b>

# Índice de figuras

---

1.1.	Producción de energías en México durante el año 2017 (Rodríguez et al., 2017)	1
1.2.	Inversión en energías renovables en billones de dolares a nivel global durante el año 2017 (Eckhart et al., 2018).	6
1.3.	Producción de energía eólica entre 2012 y 2017	6
2.1.	Representación de pronóstico a un paso y múltiples pasos en el tiempo	10
2.2.	Ejemplo de Serie de Tiempo de la velocidad del viento.	11
2.3.	Ejemplo gráfica mostrando tamaño de ventana y pronóstico a un paso en el tiempo.	11
2.4.	Ejemplo gráfica mostrando tamaño de ventana y pronóstico de múltiples pasos en el tiempo.	12
2.5.	Estrategias de pronóstico simples y compuestas	13
2.6.	Comparación de modelo Single Output con modelo Multiple Output	13
2.7.	Estrategia Recursiva para Pronóstico de Series de Tiempo para $H$ pasos a futuro.	15
2.8.	Estrategia Directa para pronósticos de series de tiempo	17
2.9.	Estructura de Estrategia MIMO para pronósticos de series de tiempo para $H$ horizonte	18
2.10.	Estructura de Estrategia DirREC	19
2.11.	Diagrama de la estructura de la Estrategia DIRMO	20
2.12.	Comparación de Inteligencia Artificial, Aprendizaje Máquina y Aprendizaje Profundo.	22
2.13.	Tipos de Aprendizaje Máquina y ejemplos de aplicaciones.	23
2.14.	Representación básica de $k$ Vecinos más Cercanos	24
2.15.	Representación cálculo distancia para $k$ Vecinos más Cercanos	25
2.16.	Estructura básica de un perceptrón	26
2.17.	Diagrama perceptrón multicapas básico para pronóstico de la velocidad del viento	27
2.18.	Diagrama de funcionamiento de una celda de las Redes de Gran Memoria a Corto Plazo	28
2.19.	Diagrama de funcionamiento de una Red Neuronal Convolucionada	29
2.20.	Ejemplo de proceso de convolución, donde se muestra multiplicación de entradas con kernel para la obtención de la salida de mapa de características.	30
2.21.	Ejemplo de los tipos de pooling: a) Max Pooling y b) Average Pooling	31
2.22.	Diagrama de proceso básico para un algoritmo genético	33
2.23.	Ejemplo simple de un individuo con genes binarios	33

2.24. Ejemplos de cruce de individuos (a) y mutación de individuo (b) . . . . .	34
2.25. Importancia de hiperparámetros para Búsqueda de Cuadrícula (GS) y Búsqueda Aleatoria (RS) . . . . .	35
3.1. Diagrama de árbol de clasificación de modelos de pronóstico (datos tomados de (Soman et al. 2010) y (Mao & Shaoshuai 2016)) . . . . .	39
4.1. Pasos seguidos por la metodología propuesta . . . . .	44
4.2. Diagrama del proceso para generar un modelo optimizado para pronosticar series de tiempo de la velocidad del viento . . . . .	45
4.3. Diagrama de preprocesamiento de datos . . . . .	47
4.4. Proceso visual de normalización de datos para una serie de tiempo . . . . .	48
4.5. Gráfica de muestra de media móvil en una serie de tiempo . . . . .	49
4.6. Gráfica de muestreo de una serie de tiempo para extraer secuencia y dato de horizonte . . . . .	50
4.7. Ejemplo de generación de secuencias a partir de una serie de tiempo . . . . .	50
4.8. Ejemplo de Serie de Tiempo a Imagen tamaño de ventana 64 . . . . .	52
4.9. Proceso visual que realiza un pronosticador con estrategia Recursiva . . . . .	54
4.10. Proceso visual que realiza un pronosticador con estrategia Directa . . . . .	55
4.11. Proceso visual que realiza un pronosticador con estrategia MIMO . . . . .	56
4.12. Proceso visual que realiza un pronosticador con estrategia DIRMOMO . . . . .	57
4.13. Proceso visual que realiza un modelo con la estrategia DirREC . . . . .	58
4.14. Combinación de modelos de Inteligencia Artificial utilizados para realizar pronóstico de la velocidad del viento . . . . .	59
4.15. Representación visual de la configuración para el modelo implementado con perceptrón multicapas (MLP) . . . . .	62
4.16. Representación visual de la configuración para el modelo implementado con redes de gran memoria a corto plazo (LSTM) . . . . .	63
4.17. Representación visual de la configuración para el modelo implementado con redes de gran memoria a corto plazo y perceptrón multicapas (LSTM-MLP) . . . . .	64
4.18. Representación visual de la configuración para el modelo implementado redes neuronales convolucionales (CNN) . . . . .	65
4.19. Representación visual de la configuración para el modelo implementado redes neuronales convolucionales y redes de gran memoria a corto plazo (CNN-LSTM) . . . . .	66
4.20. Representación visual de la configuración para el modelo implementado redes neuronales convolucionales, redes de gran memoria a corto plazo y perceptrón multicapa (CNN-LSTM-MLP) . . . . .	67
4.21. Esquema de algoritmo genético implementado . . . . .	69
5.1. Gráfica de pronóstico de 24 horas para la velocidad del viento para la serie de tiempo de Aristeo Mercado utilizando LSTM con estrategia MIMO . . . . .	75
5.2. Gráfica de pronóstico de 24 horas para la velocidad del viento para la serie de tiempo de Cointzio utilizando LSTM con estrategia MIMO . . . . .	77
5.3. Gráfica de pronóstico de 24 horas para la velocidad del viento para la serie de tiempo de Corrales utilizando LSTM con estrategia MIMO . . . . .	78

5.4.	Gráfica de pronóstico de 24 horas para la velocidad del viento para la serie de tiempo de La Piedad utilizando LSTM-MLP con estrategia DIRMO . . . .	80
5.5.	Gráfica de pronóstico de 24 horas para la velocidad del viento para la serie de tiempo de Melchor Campo utilizando LSTM con estrategia DIRMO . . . .	82

# Índice de tablas

---

2.1.	Clasificaciones de pronóstico por su rango de tiempo (Chang, 2014) . . . . .	9
2.2.	Características de estrategias de pronóstico . . . . .	14
3.1.	Distintas clasificaciones por distintos criterios para métodos de pronóstico de viento. . . . .	39
3.2.	Tabla de comparación de acercamientos para realizar pronósticos de la velocidad del viento descrita por Soman et al., 2010. . . . .	42
4.1.	Ejemplo de generación de secuencias . . . . .	50
4.2.	Tabla de ejemplo de generación de Imágenes a partir de secuencias de serie de tiempo . . . . .	52
4.3.	Hiperparámetros utilizados para la configuración del modelo MLP implementado . . . . .	62
4.4.	Hiperparámetros utilizados para la configuración del modelo LSTM implementado . . . . .	63
4.5.	Hiperparámetros utilizados para la configuración del modelo LSTM-MLP implementado . . . . .	64
4.6.	Hiperparámetros utilizados para la configuración del modelo CNN implementado . . . . .	65
4.7.	Hiperparámetros utilizados para la configuración del modelo CNN-LSTM implementado . . . . .	66
4.8.	Hiperparámetros utilizados para la configuración del modelo CNN-LSTM-MLP implementado . . . . .	68
4.9.	Parámetros utilizados para algoritmo genético . . . . .	69
5.1.	Cantidad de datos utilizados por cada una de las series de tiempo para entrenamiento y validación . . . . .	73
5.2.	Resultados obtenidos con los distintos modelos y estrategias utilizando la serie de tiempo de Aristeo Mercado . . . . .	74
5.3.	Comparación resultado de mejor resultado obtenido durante experimentos y su optimización por AG . . . . .	74
5.4.	Resultados obtenidos con los distintos modelos y estrategias utilizando la serie de tiempo de Cointzio . . . . .	76
5.5.	Comparación resultado de mejor resultado obtenido durante experimentos y su optimización por AG . . . . .	76

5.6. Resultados obtenidos con los distintos modelos y estrategias utilizando la serie de tiempo de Corrales . . . . .	78
5.7. Comparación resultado de mejor resultado obtenido durante experimentos y su optimización por AG . . . . .	78
5.8. Resultados obtenidos con los distintos modelos y estrategias utilizando la serie de tiempo de La Piedad . . . . .	79
5.9. Comparación resultado de mejor resultado obtenido durante experimentos y su optimización por AG . . . . .	79
5.10. Resultados obtenidos con los distintos modelos y estrategias utilizando la serie de tiempo de Melchor Campo . . . . .	81
5.11. Comparación resultado de mejor resultado obtenido durante experimentos y su optimización por AG . . . . .	81

---

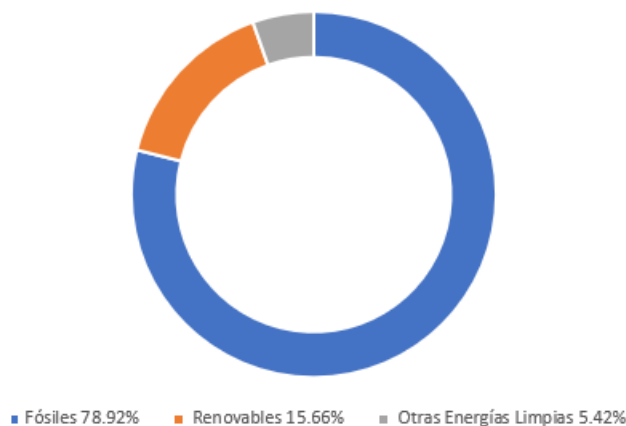
# Capítulo 1

## Introducción

---

En la actualidad la energía eléctrica convencional consumida por la población es de alta calidad y fácil de transmitir en áreas extensas. Sin embargo, la mayor parte de esta energía es generada a través del uso de recursos no renovables, tales como carbón, petróleo y gas natural. Que al momento de producir la energía emiten gases contaminantes. Una alternativa para producir energía eléctrica es utilizar la energía nuclear por fisión (Abas et al., 2015). La producción de energía nuclear es atractiva ya que no genera gases contaminantes; pero, el proceso genera residuos que pueden afectar el ambiente, por lo que se requiere almacenar de forma segura (Ewing, 2015).

En México de acuerdo a Rodríguez et al., durante el año 2017, el porcentaje de generación de energías limpias fue del 21.08 % y comparado con el 78.92 % producido por medio de recursos fósiles (Figura 1.1). Del 21.08 % de energía generada solamente el 3.23 % es energía eólica, 9.68 % hidroeléctrica, 1.84 geotérmica, 3.31 % nuclear y 3.02 % otras.



**Figura 1.1:** Producción de energías en México durante el año 2017 (Rodríguez et al., 2017)



Las tendencias globales modernas muestran cada vez mayor interés en la producción de energías limpias, durante 2017 la inversión en energías limpias fue del 68.2 %, 22.6 % para producción por medios fósiles de y 9.2 % por medios nucleares (Eckhart et al., 2018). A pesar del incremento de inversión en tecnologías para producción de energías limpias, esta inversión no es suficiente para tener un gran impacto.

La energía eléctrica producida en granjas de viento es una alternativa atractiva por tener un costo de operación bajo, además de utilizar un recurso renovable (Kumar et al., 2016). Sin embargo, el viento no es un recurso que se puede considerar constante a través del tiempo. Debido a este comportamiento del viento, el costo de producción de esta energía limpia incrementa considerablemente, dado el respaldo de energía necesario. Es por eso, la necesidad de estimar la cantidad de materia prima a disponer para estimar la cantidad de energía a producir. Un pronóstico más preciso reducirá considerablemente los costos de producción de este tipo de energía.

Este trabajo aborda el problema del pronóstico de viento a corto plazo, siendo el intervalo de predicción las próximas 24 horas en escala horaria. Para lograr este objetivo se utilizan técnicas de Inteligencia Artificial (IA), particularmente los modelos de Redes Neuronales Artificiales (ANN): Perceptron Multicapas (MLP), Redes Neuronales Convolucionales (CNN) y Redes de Gran Memoria a Corto Plazo (LSTM).

El uso de Inteligencia Artificial (IA) ha crecido en popularidad en los últimos años para realizar pronósticos, ya que han demostrado obtener mejores resultados que los métodos clásicos (i.e. Auto regresivo, Auto regresivo con Media Móvil, etc.) para pronósticos con horizontes amplios (Soman et al., 2010). Los modelos clásicos destacan al pronosticar variables a un solo paso en el tiempo, mientras que los modelos de IA obtienen resultados con buena aptitud para pronósticos a múltiples pasos en el tiempo.

La tarea de realizar pronósticos a múltiples pasos en el tiempo no es una tarea sencilla. Es por esto, que es necesario utilizar diferentes estrategias de pronóstico a múltiples pasos (Taieb, Bontempi, Atiya et al., 2012).

Para este trabajo se utilizan 5 diferentes estrategias para pronósticos a múltiples pasos: Estrategia Recursiva, Estrategia Directa, Estrategia Múltiples Entradas Múltiples Salidas (MIMO), Estrategia DirREC (Directa-Recursiva) y Estrategia DIRMO (Directa-MIMO).

Este trabajo muestra los distintos modelos de Inteligencia Artificial combinados con las distintas estrategias de pronósticos, donde se muestran las fortalezas y desventajas que existen en cada una de estas combinaciones.

El proceso de entrenamiento realizado en este trabajo requiere un preprocesamiento a las series de tiempo que se utilizaran para entrenar los modelos pronosticadores ya que pueden existir datos atípicos por fallas en sensores, captura de datos, u otras anomalías físicas (Montgomery et al., 2015). Además, la serie de tiempo se transforma a secuencias de datos para los modelos Perceptron Multicapas y Redes de Gran Memoria a Corto Plazo, y a imágenes utilizando la técnica de *Grammian Angular Field* (GAF) para las Redes Neuronales Convolucionales.

Inicialmente, la optimización de los modelos predictivos (redes neuronales) generados, se realizó mediante prueba y error, obteniendo errores considerables. Es por esta razón que se utilizan técnicas de optimización de hiperparámetros para automatizar esta tarea (Bergstra et al., 2011). Las técnicas utilizadas en este trabajo son los Algoritmos Genéticos (AG) y Búsqueda Aleatoria (RS), donde se realizó una inicialización de población para AG por RS.

La metodología propuesta en este trabajo muestra el procedimiento seguido para realizar pronósticos a 24 horas en el futuro de la velocidad del viento. Se utilizó un conjunto de métodos de Inteligencia Artificial y estrategias para pronósticos, utilizando técnicas de optimización de hiperparámetros buscando obtener resultados con una mejor aptitud.

## **1.1. Definición del problema**

La demanda global de energía crece cada año debido al incremento de la población e incremento de automatización de actividades cotidianas. Durante el año 2017 la demanda por energía creció en 2.1 %, mas del doble de los 5 años previos (Eckhart et al., 2018). Sin embargo, el verdadero problema se encuentra en su forma de producción. Un estimado de 73.5 % de la energía es producida por medios fósiles, mientras que solamente 26.5 % son energías limpias modernas (Eckhart et al., 2018). En México entre el año 2004 y 2017 mas de tres cuartos de la energía producida para abastecimiento de la población se ha generado utilizando medios fósiles (Rodríguez et al., 2017). El incremento de consumo de recursos

fósiles tales como: petróleo, gas y carbón, tiene como consecuencia el aumento de niveles de contaminación.

Una de las alternativas para la producción, es la energía eólica, la cual es limpia, viable y en creciente popularidad (Kumar et al., 2016). El interés para desarrollo tecnológico utilizado en la producción de energía limpia aumenta año con año; por lo cual, aumenta la viabilidad y se reducen costos de producción de energía eólica. Durante el año 2017 la producción de energía eólica en México se incremento por 12.41 % (Rodríguez et al. ,2017).

La producción de energía eólica presenta varios desafíos, entre ellos el conocer con antelación la velocidad de viento que estará disponible en las horas futuras. Esto permite realizar ajustes en la planta de producción de energía eólica, reduciendo costos de producción y conocer la cantidad de producción energética que será posible producir (Santamaría-Bonfil et al., 2016). Es por esto que surge la necesidad de pronosticar el comportamiento del viento para ayudar la creciente adopción por el uso de energías eólicas.

## **1.2. Hipótesis**

El uso de distintos modelos de Inteligencia Artificial (IA) combinados con diferentes técnicas de pronóstico a múltiples pasos obtendrán pronósticos con distintos niveles aptitud, ya que podrán combinar distintas fortalezas existentes entre los modelos de IA y las técnicas de pronóstico; lo cual hace posible la selección de un modelo superior.

## **1.3. Objetivo**

Desarrollar una metodología para obtener un modelo capaz de pronosticar la velocidad del viento a corto plazo (24 horas) utilizando técnicas de Inteligencia Artificial, estrategias de pronóstico a múltiples pasos y técnicas de optimización de hiperparámetros.

## 1.4. Objetivos particulares

1. Diseñar la metodología utilizada para generar modelos pronosticadores para la velocidad del viento a múltiples pasos en el tiempo (24 horas).
2. Realizar experimentos de pronósticos para la velocidad del viento utilizando los modelos generados con distintas series de tiempo.
3. Comparar los distintos resultados obtenidos por los múltiples modelos de IA y estrategias de pronóstico a múltiples pasos.
4. Utilizar técnicas de optimización de hiperparámetros automatizar la búsqueda de hiperparámetros que mejoren la aptitud del modelo pronosticador.

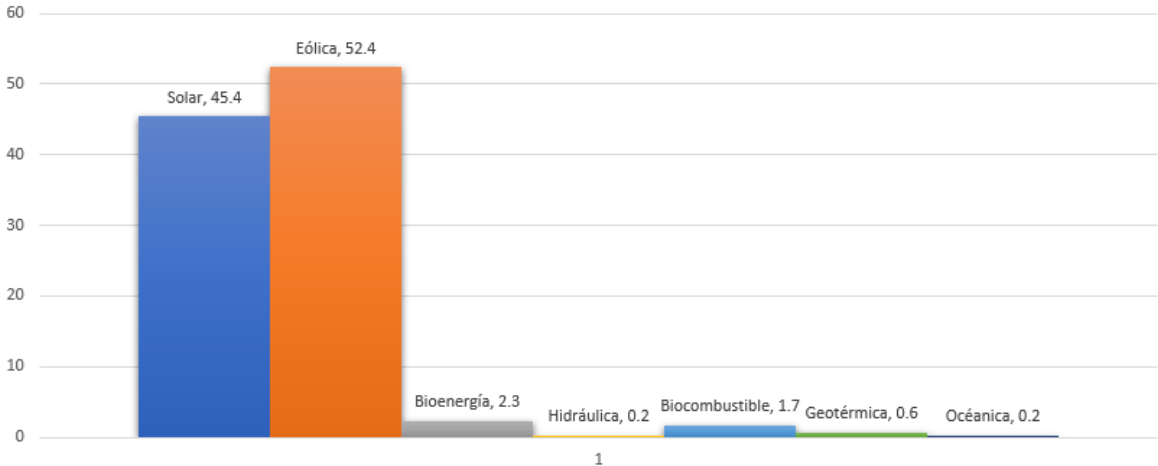
## 1.5. Justificación

El uso de modelos para pronósticos de la velocidad del viento a 24 horas tienen la capacidad de ayudar a conocer el comportamiento del viento que se tendrá en las granjas eólicas. Esto ayudará a reducir los costos de producción de energía al permitir conocer la capacidad energética con antelación y realizar los ajustes necesarios, tales como programación de mantenimientos, ajustes en las turbinas generadoras, generación de energía por medios alternativos, entre otros.

También, conocer el comportamiento que tendrá el viento con antelación, permite determinar, entre otros factores, la operabilidad de una turbina eólica; ya que, si la velocidad del viento es muy baja, la energía producida no cubrirá el costo de producción, y si la velocidad es muy alta se deberá detener la operación ya que se pueden generar daños si la turbina eólica no está diseñada para esas velocidades (Kumar et al., 2016).

En los últimos años la inversión para producción de energías limpias han tenido un crecimiento notable, esto se debe a la evolución tecnológica y viabilidad que presentan las tecnologías para producir energías limpias (Eckhart et al., 2018). Durante el año 2017 la inversión económica global para el desarrollo de la producción de energía eólica fue de 52.4 billones de dolares, seguida por la energía solar por 45.4 billones y el resto de las formas de producción

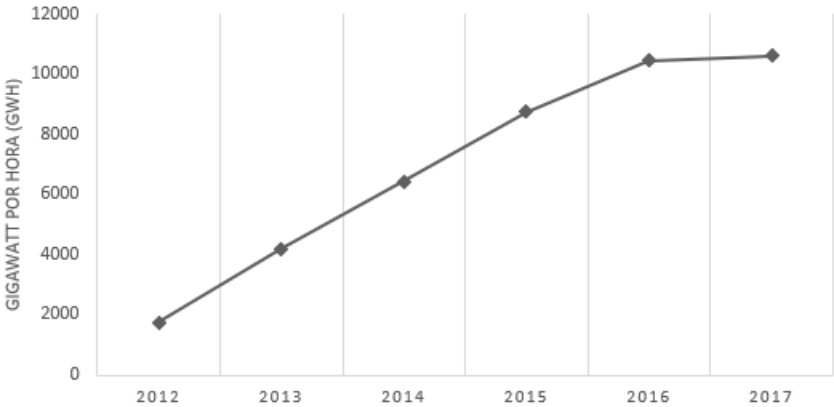
de energías limpias, como se puede observar en la Figura 1.2. La producción de energía por medio del viento ha crecido gracias a la evolución tecnológica que permite mayor producción a un menor costo; esto se debe al incremento en la inversión para producción de energías limpias debido al aumento de costo de recursos fósiles y expectativas futuras de su extracción (Abas et al., 2015).



**Figura 1.2:** Inversión en energías renovables en billones de dolares a nivel global durante el año 2017 (Eckhart et al., 2018).

Entre el año 2012 y 2017 la generación de energía eólica mostró un crecimiento anual promedio del 43.5 %, como se muestra en la Figura 1.3 (Rodríguez et al. 2017). Se espera para el año 2021 tener un aumento del 91 % en la producción de energía eólica.

El incremento a la inversión en energías limpias tal como la energía eólica tienen gran



**Figura 1.3:** Producción de energía eólica entre 2012 y 2017

impacto climatológico al reducir la cantidad de contaminación ambiental generada al utilizar recursos fósiles, al mismo tiempo que ayuda a producir energía para la población (Bhattacharya et al., 2016). Es por esto que ha surgido la necesidad de conocer la cantidad de materia prima que se tendrá con antelación para facilitar la tarea de producción de energía eólica. Con el objetivo de contribuir a la reducción de costos de producción, motivando la completa adopción de energías limpias.

Los modelos basados en Inteligencia Artificial ha demostrado poder realizar pronósticos de la velocidad del viento, los cuales han beneficiado granjas eólicas al permitir conocer el comportamiento del viento con antelación (Santamaría-Bonfil et al., 2016). Además, su creciente popularidad y facilidad de uso las han convertido en una de las primeras opciones para la solución de problemas de regresión. Es por esto que para este trabajo se propone el uso de distintos modelos de Inteligencia Artificial, con el objetivo de buscar utilizar sus fortalezas y obtener modelos pronosticadores que permitan lograr conocer con un bajo error la velocidad del viento.

## **1.6. Estructura de la tesis**

Este trabajo consta de 6 capítulos que se encuentran organizados de la siguiente manera. El capítulo 2 explica la teoría fundamental requerida para comprender el contenido de este trabajo, el capítulo 3 habla de trabajos similares para técnicas y tecnologías utilizadas en este trabajo, en el capítulo 4 se explican los pasos realizados para obtener un modelo pronosticador, el capítulo 5 muestra algunos de los resultados obtenidos con las distintas técnicas, modelos y series de tiempo, finalmente el capítulo 6 habla de las conclusiones de este trabajo además de comentar trabajos futuros.

---

# Capítulo 2

## Marco teórico

---

Este capítulo aborda los conceptos teóricos utilizados durante el desarrollo de este trabajo. Se comienza explicando conceptos básicos tales como energía eólica y series de tiempo. Posteriormente se exponen los conceptos teóricos para las distintas estrategias de pronósticos y modelos de Inteligencia Artificial utilizados durante la elaboración del proyecto. Para continuar se explican las técnicas de optimización de hiperparámetros utilizados al ajustar de forma automática los modelos. Finalmente se muestran las distintas métricas estadísticas utilizadas para calcular el error durante la evaluación y entrenamiento de los modelos.

### 2.1. Energía eólica

El viento es el recurso utilizado para la producción de energía eólica. La fuente original que genera el viento es el sol, esto se debe a que el viento es creado por el calentamiento desigual sobre la superficie de la tierra, debido a la superficie irregular que posee el planeta (Manwell et al., 2010). Creando diferencias de presión en el aire, forzando a moverse de zonas de alta a baja presión.

La energía eólica se genera cuando el viento genera un movimiento mecánico, el cual se transfiere a un aerogenerador que produce energía eléctrica (Kumar et al., 2016). La producción de energía eólica tiene la ventaja de no producir contaminación en el aire además de utilizar un recurso que es renovable (Dell & Rand, 2004); es por esta razón que tiene el potencial de ser una alternativa limpia con el potencial de cubrir la creciente necesidad de consumo energético global sin afectar el medio ambiente (Quaschnig, 2019).

## 2.2. Pronóstico

Un pronóstico es una predicción de uno o varios eventos futuros (Montgomery et al., 2015). Este es uno de los desafíos que se puede encontrar en múltiples campos, como lo son: la industria, medicina, economía, meteorología, política, etc. La importancia de los pronósticos se debe a la ventaja que puede proveer el conocer con antelación distintos sucesos, por ejemplo, el conocer el comportamiento de la bolsa de valores, predecir la falla de un equipo industrial, conocer el clima que se tendrá el día siguiente, entre otros.

Los pronósticos se pueden clasificar de acuerdo a su rango de tiempo, una forma común de clasificarlos es: ultra corto plazo, corto plazo, medio plazo y largo plazo (Chang, 2014). En la Tabla 2.1 se muestran las distintas escalas de tiempo, el rango de tiempo y cada una de sus aplicaciones de acuerdo a la escala.

**Tabla 2.1:** Clasificaciones de pronóstico por su rango de tiempo (Chang, 2014)

Escala de Tiempo	Rango de Tiempo	Aplicaciones
Ultra Corto Plazo	Desde unos minutos hasta una hora	<ul style="list-style-type: none"><li>• Compensación del mercado eléctrico</li><li>• Operaciones de redes en tiempo real</li><li>• Reguladores</li></ul>
Corto Plazo	Desde una hasta múltiples horas adelante	<ul style="list-style-type: none"><li>• Planificación de despacho de carga económica</li><li>• Pronóstico de la velocidad del viento</li></ul>
Medio Plazo	Desde múltiples horas hasta una semana	<ul style="list-style-type: none"><li>• Decisión de encendido y apagado de generador</li><li>• Clima</li></ul>
Largo Plazo	Desde una semana hasta un año o mas	<ul style="list-style-type: none"><li>• Planeación de mantenimientos</li><li>• Estudio de viabilidad para construcción de granja eólica</li></ul>

A pesar de la gran cantidad de áreas en las que se pueden realizar pronósticos, se puede clasificar en dos tipos de técnicas para pronóstico (Montgomery et al., 2015):

- Cualitativo
- Cuantitativo

Los pronósticos cualitativos son frecuentemente subjetivos en naturaleza y requieren el juicio de un experto. Este pronóstico es frecuentemente utilizado cuando hay pocos o nulos datos con los cuales realizar un pronóstico. Un pronóstico cuantitativo por otro lado resume los patrones en los datos y expresa una relación estadística entre valores previos y presentes, posteriormente el modelo proyecta estos patrones para realizar observaciones futuras (Montgomery et al., 2015).



Los pronósticos pueden realizarse a un solo paso en el tiempo, o múltiples pasos, siendo los pronósticos a un solo paso los más simples (D. I. Harvey et al., 1998). El realizar pronósticos a múltiples pasos tienen la ventaja de permitirnos conocer un mayor *horizonte* (número de observaciones futuras a realizar) de datos futuros, sin embargo, aumenta la dificultad de pronóstico debido a distintos factores que se revisaran en la Sección de *Pronóstico de Múltiples pasos en el Tiempo*. En la Figura 2.1 se puede observar una simple representación de un pronóstico a un paso y múltiples pasos, donde  $y_t \dots y_{t-d}$  (cuadros azules) son los datos históricos utilizados para realizar el pronóstico y  $\hat{y}_{t+1} \dots \hat{y}_{t+H}$  son las observaciones futuras realizadas por el modelo pronosticador hasta  $H$  (horizonte máximo) observaciones.



**Figura 2.1:** Representación de pronóstico a un paso y múltiples pasos en el tiempo

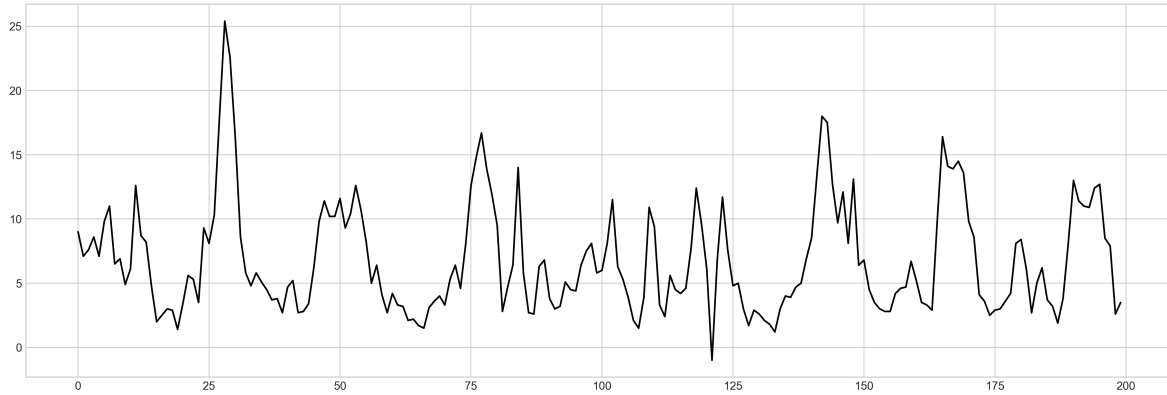
### 2.2.1. Serie de tiempo

Una serie de tiempo es una secuencia cronológica de observaciones de una variable de interés (Montgomery et al., 2015). Las series de tiempo se pueden clasificar en dos tipos principales:

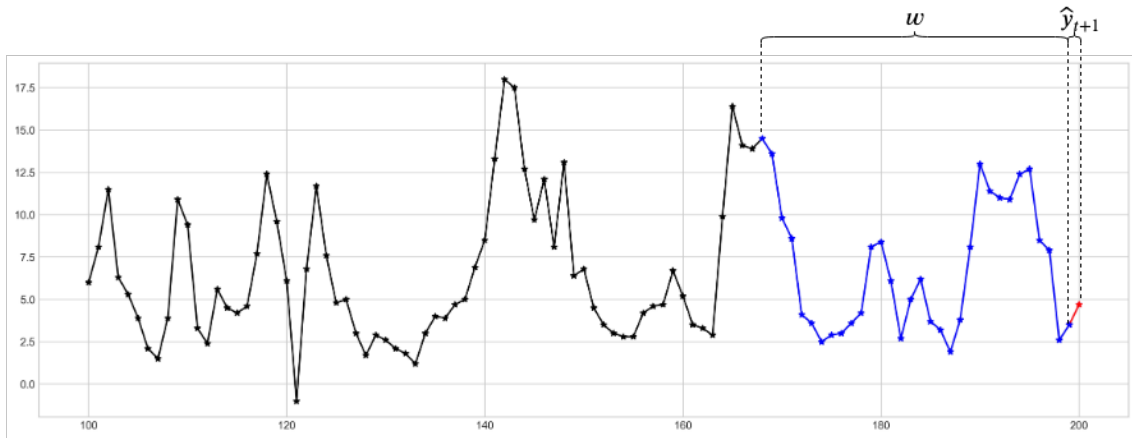
- Series de tiempo discretas
- Series de tiempo continuas

Una serie discreta toma observaciones a intervalos fijos de tiempo, mientras que una serie de tiempo continua toma valores a un intervalo infinitésimo mínimo (Brockwell et al., 2002).

Los pronósticos cuantitativos, que es el tipo de pronóstico que se aborda en este trabajo, utilizan series de tiempo discretas. Los intervalos de tiempo que se utilizan para realizar pronósticos pueden variar de acuerdo al rango de tiempo que se desea pronosticar. Estos rangos de tiempo pueden ser de segundos, minutos, horas, días, semanas, meses, años, etc. En la Figura 2.2 se puede observar un ejemplo gráfico de serie de tiempo de la velocidad del



**Figura 2.2:** Ejemplo de Serie de Tiempo de la velocidad del viento.



**Figura 2.3:** Ejemplo gráfica mostrando tamaño de ventana y pronóstico a un paso en el tiempo.

viento con escala horaria, donde cada punto individual en la gráfica representa el estado de velocidad de viento en ese termino horario.

### 2.2.2. Pronóstico a un paso en el tiempo

Los pronósticos a un paso son sencillos ya que se puede asumir que los errores de pronóstico no están autocorrelacionados (D. Harvey & Newbold, 2000). El modelo pronosticador a un paso se puede definir como:

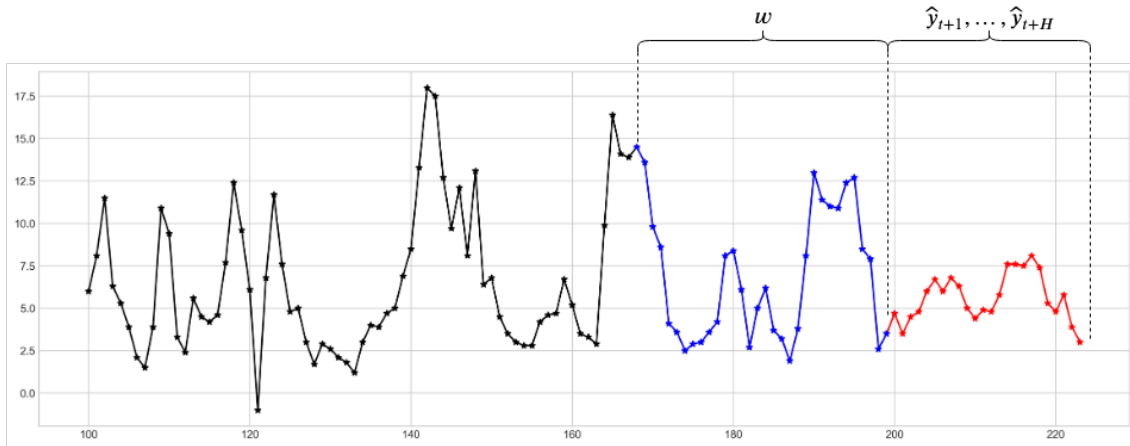
$$\hat{y}_{t+1} = f(y_t, \dots, y_{t-w+1}) \quad (2.1)$$

donde  $\hat{y}$  es el dato pronosticado,  $f$  es el modelo pronosticador,  $t$  es el paso en el tiempo del dato y  $w$  es el tamaño de ventana de datos que toma el modelo para realizar el pronóstico (Taieb, Bontempi, Atiya et al., 2012). En la Figura 2.3 se puede observar un ejemplo de serie

de tiempo, donde  $w$  es la ventana de datos de muestreo tomada para realizar el pronóstico, y  $\hat{y}_{t+1}$  es el pronóstico de un paso hacia el futuro.

### 2.2.3. Pronóstico a múltiples pasos en el tiempo

Los pronósticos a un paso en el tiempo pueden ser no suficientes para resolver problemáticas en aplicaciones reales, es por ello que se busca conocer múltiples pasos en el tiempo al realizar pronósticos. A diferencia del pronóstico a un paso en el tiempo, el pronóstico de múltiples pasos en el tiempo es considerablemente más difícil debido a la acumulación de errores, reducción de precisión, e incremento de incertidumbre (An & Anh, 2015). En la Figura 2.4 se puede observar un ejemplo de serie de tiempo, similar a la Figura 2.3, con la diferencia del tamaño de horizonte ( $H$ ) pronosticado.

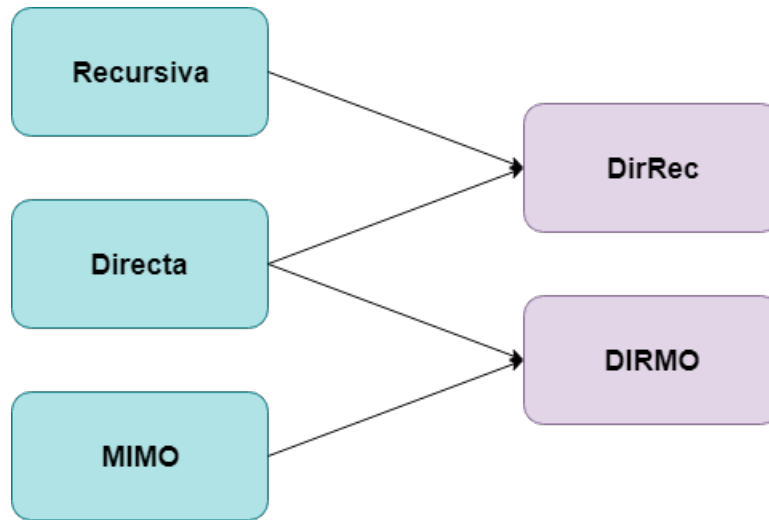


**Figura 2.4:** Ejemplo gráfica mostrando tamaño de ventana y pronóstico de múltiples pasos en el tiempo.

### 2.2.4. Estrategias para pronósticos de múltiples pasos en el tiempo

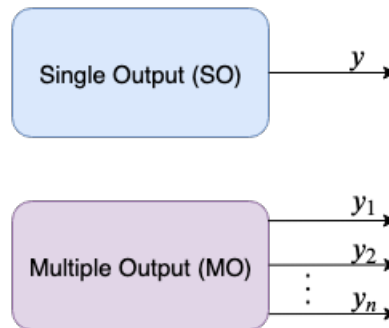
Para realizar pronósticos a múltiples pasos en el tiempo se han propuesto diferentes estrategias con la finalidad mejorar la aptitud de los pronósticos. En este trabajo se utilizan 5 estrategias de pronóstico conocidas: Recursiva, Directa, MIMO, DirREC y DIRMO (Taieb, Bontempi, Atiya et al., 2012). Las primeras tres estrategias son consideradas estrategias base, mientras que DirREC y DIRMO son consideradas estrategias compuestas. Cada estrategia

presenta ventajas y desventajas sobre las otras, tales como poder computacional requerido, precisión, complejidad, entre otros aspectos que serán revisados posteriormente en cada estrategia individual. En la Figura 2.5 se observa la relación entre las estrategias simples y compuestas; como se muestra, la estrategia DirREC es una combinación de la estrategia Directa y Recursiva, mientras que DIRMO es una combinación de la estrategia Directa y MIMO.



**Figura 2.5:** Estrategias de pronóstico simples y compuestas

Los tipos de modelos utilizados en las estrategias de pronóstico se pueden clasificar en 2 tipos: de una salida o por su nombre en inglés *Single Output* (SO) y de múltiples salidas o por su nombre en inglés *Multiple Outputs* (MO). En la Figura 2.6 se puede observar la diferencia entre un modelo SO y un modelo MO, donde SO solamente cuenta con una salida ( $y_{t+1}$ ), mientras que un modelo MO puede tener hasta  $n$  número de salidas ( $y_{t+1}, y_{t+2}, \dots, y_{t+n}$ ).



**Figura 2.6:** Comparación de modelo Single Output con modelo Multiple Output

Un aspecto importante que se debe considerar al elegir una estrategia de pronóstico es el

**Tabla 2.2:** Características de estrategias de pronóstico

Estrategia	Número de Modelos	Tipo de Modelo	Tamaño de Salida	Tiempo Computacional Requerido
Recursiva	1	SO	1	$1 \times T_{SO}$
Directa	H	SO	1	$H \times T_{SO}$
DirRec	H	SO	1	$H \times (T_{SO} + \mu)$
MIMO	1	MO	H	$1 \times T_{MO}$
DIRMO	$\frac{H}{s}$	MO	s	$\frac{H}{s} \times T_{MO}$

poder computacional para entrenar los modelos. Sea  $T_{SO}$  y  $T_{MO}$  denotaciones para la cantidad de tiempo requerido para entrenar un modelo, para modelos SO y MO respectivamente. En la Tabla 2.2 se puede observar las características para cada estrategia dado  $H$  horizonte (Taieb, Bontempi, Atiya et al., 2012). Los modelos de acuerdo a su complejidad computacional se pueden clasificar como se muestran en la Ecuación 2.2, donde  $s$  es cantidad de salidas para modelo DIRMO, suponemos que  $1 < s < H$  y las estrategias van de izquierda a derecha de menor a mayor complejidad computacional.

$$\underbrace{1 \times T_{SO}}_{\text{Recursiva}} < \underbrace{1 \times T_{MO}}_{\text{MIMO}} < \underbrace{\frac{H}{s} \times T_{MO}}_{\text{DIRMO}} < \underbrace{H \times T_{SO}}_{\text{Directa}} < \underbrace{H \times (T_{SO} + \mu)}_{\text{DirREC}} \quad (2.2)$$

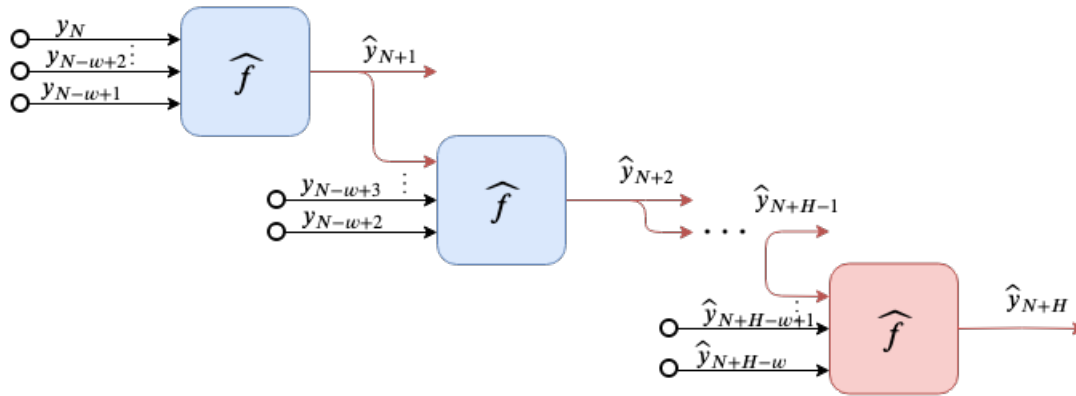
En las siguientes secciones se explicarán a detalle cada una de las 5 estrategias de pronósticos a múltiples pasos.

#### 2.2.4.1. Estrategia Recursiva

La estrategia Recursiva consiste en entrenar un modelo  $f$  para pronosticar un solo paso en el tiempo como se muestra en la Ecuación 2.3, donde  $r$  es el ruido,  $y$  la variable de interés,  $w$  el tamaño de ventana y  $t \in \{w, \dots, N - 1\}$  (donde  $N$  es el dato inicial posterior a la ventana  $w$ ) (Taieb, Bontempi, Atiya et al., 2012).

$$y_{t+1} = f(y_t, \dots, y_{t-w+1}) + r \quad (2.3)$$

Cuando se busca pronosticar  $H$  pasos en el futuro, primero se pronostica un paso aplican-



**Figura 2.7:** Estrategia Recursiva para Pronóstico de Series de Tiempo para H pasos a futuro.

do el modelo  $f$ . El valor obtenido en el primer pronóstico alimenta al modelo para pronosticar el siguiente paso en el tiempo, repitiendo el proceso hasta predecir el horizonte  $H$  completo (An & Anh, 2015). En la Figura 2.7 se puede observar la arquitectura de la estrategia Recursiva.

En el conjunto de Ecuaciones (2.4) se observa que  $\hat{y}_{t+1}$  se estima a partir de entradas de la serie y valores estimados. Cuando  $h = 1$ , el modelo solo usa valores de la serie para pronosticar un solo paso. Si  $2 \geq h \geq d$ , el modelo se alimenta de valores estimados y de la serie de tiempo; finalmente, cuando  $h > w$ , el modelo se alimenta únicamente de valores pronosticados.

$$\hat{y}_{N+h} = \begin{cases} \text{If } h = 1 & \hat{f}(y_N, \dots, y_{N-w}) \\ \text{If } h \in (2, \dots, w) & \hat{f}(\hat{y}_{N+h-1}, \dots, \hat{y}_{N+1}, \dots, y_{N-w+h}) \\ \text{If } h \in (w+1, \dots, H) & \hat{f}(\hat{y}_{N+h-1}, \dots, \hat{y}_{N+h-w}) \end{cases} \quad (2.4)$$

La Estrategia Recursiva sufre de la acumulación de errores por cada paso que recorre para  $h > 1$  hasta llegar al horizonte máximo  $H$ . Esto es especialmente cierto cuando  $h > w$ , como plantea el tercer caso de el conjunto de Ecuaciones (2.4), donde el modelo comienza a ser alimentado con solamente valores previamente pronosticados (Taieb, Bontempi, Atiya et al., 2012).

Las ventajas (+) y desventajas (-) de la estrategia recursiva son las siguientes:

- + Estrategia más rápida para su entrenamiento.

- + Simple de programar.
- Tiene acumulación de errores que crece a mayor horizonte.

A pesar la desventaja de la acumulación de errores la Estrategia Recursiva ha demostrado obtener resultados favorables con redes neuronales recurrentes (Saad et al., 1998) y con algoritmo de k vecinos más cercanos (Bontempi et al., 2012).

#### 2.2.4.2. Estrategia Directa

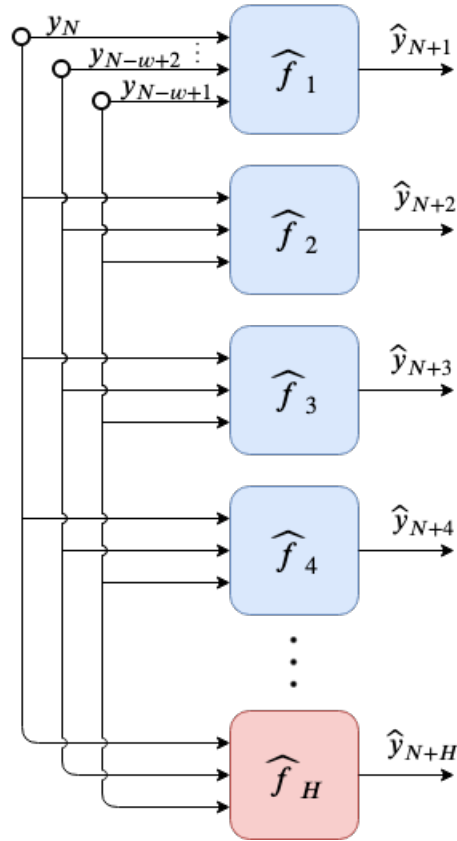
La estrategia Directa se caracteriza por pronosticar cada paso del horizonte en tiempo de manera independiente (Taieb, Bontempi, Atiya et al., 2012). Es decir, si se tiene un horizonte  $H$ , se debe entrenar la misma cantidad de modelos para cada paso de tiempo que conforma el horizonte completo. La Ecuación 2.5 representa la estrategia Directa, donde  $t \in (w, \dots, N - H)$ ,  $N$  representa el punto inicial de datos de validación,  $H$  representa el horizonte de pronóstico máximo y  $h \in (1, \dots, H)$ .

$$y_{t+h} = f_h(y_t, \dots, y_{t-w+1}) + r \quad (2.5)$$

La estrategia Directa presenta la ventaja de no utilizar valores aproximados ya que no retroalimenta datos anteriormente pronosticados, además de poseer la capacidad de ajustar cada modelo pronosticador individualmente. La Figura 2.8 muestra la estructura utilizada para la estrategia donde se puede observar que cada bloque representa un modelo individual.

A pesar de las ventajas que presenta la estrategia Directa, el costo computacional requerido para el entrenamiento individual y el ajuste de los parámetros de cada modelo es una gran desventaja, por lo que se debe considerar al comparar con otras estrategias. La Estrategia Directa presenta las siguientes ventajas (+) y desventajas (-):

- + No presenta acumulación de errores.
- + Cada modelo puede ser configurado de forma independiente.
- Puede generar independencia entre las variables pronosticadas.
- El entrenar múltiples modelos aumenta el poder computacional requerido.



**Figura 2.8:** Estrategia Directa para pronósticos de series de tiempo

La Estrategia Directa ha sido utilizada en múltiples algoritmos de aprendizaje máquina, los cuales han demostrado obtener resultados que logran satisfacer criterios requeridos en aplicaciones de negocios (Kline, 2004).

### 2.2.4.3. Estrategia MIMO

La estrategia MIMO (Multiple Input - Multiple Output) posee múltiples salidas, cada una de las cuales corresponde a un horizonte distinto. Por lo anterior, la estrategia MIMO emplea solamente un modelo  $F$  para la serie de tiempo  $[y_t, \dots, y_N]$  como se muestra en la Ecuación 2.6, donde  $t \in \{w, \dots, N - H\}$ ,  $F : \mathbb{R}^w \rightarrow \mathbb{R}^H$  es un modelo vectorial y  $r \in \mathbb{R}^H$  es un vector de ruido (Taieb, Bontempi, Atiya et al., 2012).

$$[y_H, \dots, y_{t+1}] = F(y_t, \dots, y_{t-w+1}) + r \quad (2.6)$$

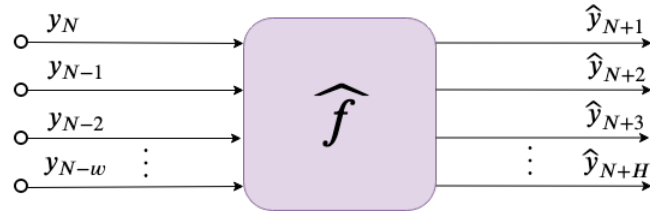


El pronóstico obtenido por un modelo utilizando la Estrategia MIMO se expresa por la Ecuación 2.7, donde  $N$  es el paso de tiempo del dato y  $w$  es el tamaño de ventana utilizado.

$$[\hat{y}_{t+H}, \dots, \hat{y}_{t+1} = \hat{F}(y_N, \dots, y_{N-w+1})] \quad (2.7)$$

El ajuste y calibración realizados en un modelo MIMO juegan un papel importante para modelos pronosticadores (Taieb, Bontempi, Sorjamaa et al., 2009) Además, ajustar un solo modelo MIMO requiere menor poder computacional comparado con un modelo que utiliza la estrategia Directa.

La Figura 2.9 muestra la estructura utilizada por la estrategia MIMO, donde se presenta un solo modelo con múltiples entradas y salidas.



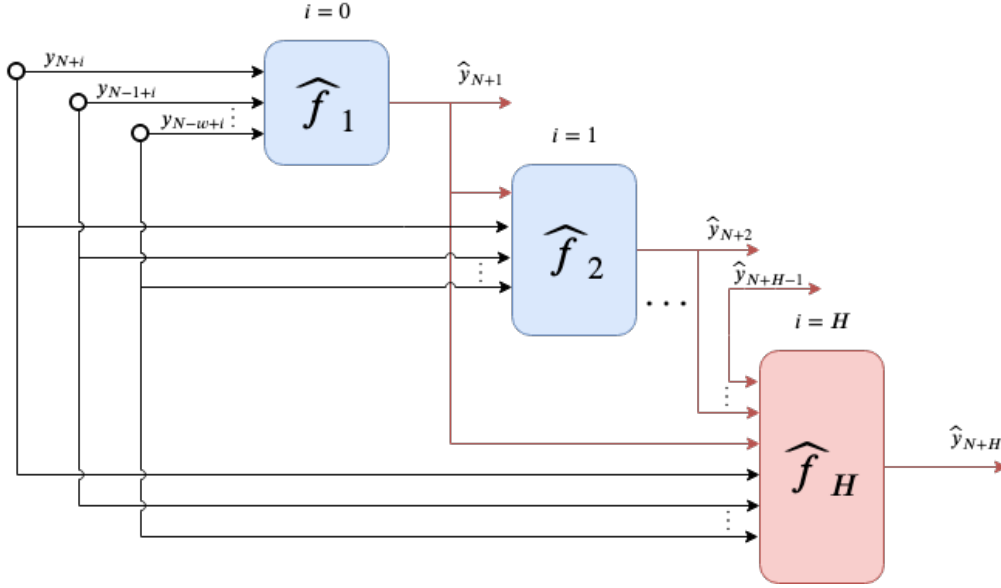
**Figura 2.9:** Estructura de Estrategia MIMO para pronósticos de series de tiempo para  $H$  horizonte

Las ventajas (+) y desventajas (-) que presenta un modelo que utiliza la Estrategia MIMO son:

- + No genera independencia entre las variables pronosticadas.
- + Bajo poder computacional requerido.
- Tiene menor flexibilidad que otras estrategias.

#### 2.2.4.4. Estrategia DirREC

La estrategia DirREC (Figura 2.10) combina las estructuras y principios de las estrategias Directa y Recursiva (An & Anh, 2015). La estrategia DirREC calcula cada pronóstico del paso en el tiempo de forma individual, tal como lo realiza la estrategia Directa. Además, agrega a su entrada cada pronóstico realizado por el modelo del paso anterior de tiempo, como lo realiza la estrategia Recursiva. La estrategia Directa entrena  $H$  modelos  $f_h$  con la



**Figura 2.10:** Estructura de Estrategia DirREC

serie  $[y_1, \dots, y_N]$  como muestra la Ecuación 2.8, donde  $t \in \{w, \dots, N-H\}$  y  $h \in \{1, \dots, H\}$  (Taieb, Bontempi, Atiya et al., 2012).

$$y_{t+h} = f_h(y_{t+h-1}, \dots, y_{t-w+1}) + r \quad (2.8)$$

El conjunto de Ecuaciones 2.9 muestra la condición necesaria para la implementación de la estrategia DirREC, donde el horizonte de pronóstico debe ser mayor a un paso.

$$\hat{y}_{N+h} = \begin{cases} \text{If } h = 1 & \hat{f}_h(y_N, \dots, y_{N-w+1}) \\ \text{If } h \in (2, \dots, H) & \hat{f}_h(\hat{y}_{N+h-1}, \dots, \hat{y}_{N+1}, \hat{y}_N, \dots, y_{N-w+1}) \end{cases} \quad (2.9)$$

La estrategia DirREC presenta las siguientes ventajas (+) y desventajas (-):

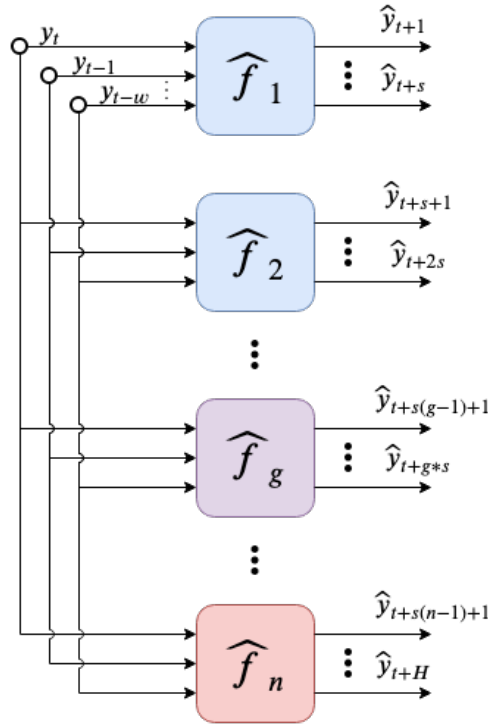
- + Puede intercambiar entre las ventajas y desventajas que presentan la estrategia Directa y estrategia Recursiva.
- La cantidad de entradas que posee cada modelo  $H$  crece de manera lineal al agregarse la retroalimentación del modelo anterior.
- Requiere el mayor poder computacional de las estrategias.

### 2.2.4.5. Estrategia DIRMO

La estrategia DIRMO tiene como objetivo preservar las mejores cualidades de la estrategia Directa y estrategia MIMO. Esta estrategia busca pronosticar  $H$  tamaño de horizonte en  $s$  cantidad de bloques, donde cada bloque es un modelo independiente que pronostica  $n$  salidas (Taieb, Bontempi, Atiya et al., 2012). La cantidad de salidas que tendrá cada bloque esta dada por la Ecuación 2.10, donde  $s \in \{1, \dots, H\}$ .

$$n = \frac{H}{s} \quad (2.10)$$

El parámetro  $s$  permite la flexibilidad de calibrar la cantidad de salidas de cada modelo pronosticador o la cantidad de modelos pronosticadores. En la Figura 2.11 se muestra la estructura de la estrategia DIRMO.



**Figura 2.11:** Diagrama de la estructura de la Estrategia DIRMO

La estrategia DIRMO previamente conocida como MISMO (Taieb, Bontempi, Sorjamaa et al., 2009), entrena  $n$  modelos  $F_p$  para una serie de tiempo  $[y_1 \dots, y_N]$ . La Ecuación 2.11 define la estrategia DIRMO, donde  $t \in \{w, \dots, N - H\}$ ,  $p \in \{1, \dots, n\}$  y  $\mathbb{R}^w \rightarrow \mathbb{R}^s$  es una

función vectorial si  $s < 1$  (Taieb, Bontempi, Atiya et al., 2012).

$$[y_{t+p*s}, \dots, y_{t+(p-1)*s+1}] = F_p(y_t, \dots, y_{t-w+1}) + r \quad (2.11)$$

La Estrategia DIRMO presenta las ventajas (+) y desventajas (-) siguientes:

- + Capacidad de intercambiar entre una mayor cantidad de salidas por modelo o número de modelos lo que aumenta la flexibilidad.
- Se requiere un nuevo parámetro  $s$  para definir cantidad de bloques.
- Poder computacional puede aumentar o disminuir de acuerdo al parámetro  $s$ .

## 2.3. Inteligencia Artificial

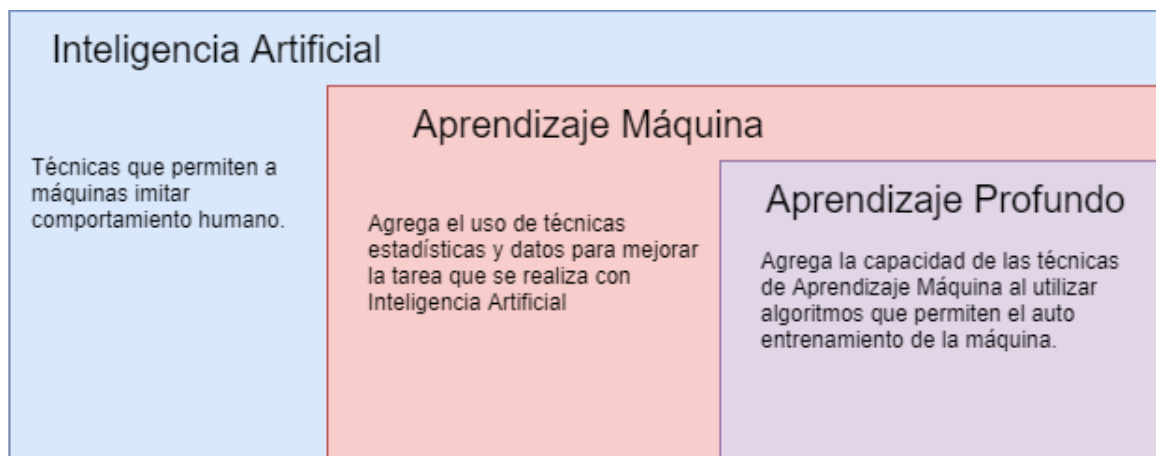
La Inteligencia Artificial (IA) es una ciencia que define como meta hacer a las máquinas capaces de realizar tareas humanas (Michael, 2005). Sin embargo, se llegó a la conclusión que el dominio del problema para las máquinas inteligentes debía ser restringido de sistemas de propósito general a sistemas que resuelven un problema específico. Otra definición para la IA por McCarthy (McCarthy, 2004) es: *”la IA es la ciencia e ingeniería dedicada a la creación de máquinas inteligentes, especialmente programas de computadora inteligentes. Esta relacionada a tareas similares de utilizar computadoras para comprender la inteligencia humana, pero la IA en si misma no está restringida solamente a métodos que son biológicamente observables”*.

Los sistemas de IA se especializan de acuerdo a las herramientas y técnicas que se emplean. Estas especializaciones son:

- Aprendizaje Máquina
- Aprendizaje Profundo

donde Aprendizaje Máquina o *Machine Learning* (ML) por su nombre en inglés, es una especialización de IA y Aprendizaje Profundo o *Deep Learning* (DL) por su nombre en

inglés, es una especialización de Aprendizaje Máquina. En la Figura 2.12 se puede observar una representación de la especialización de IA a Aprendizaje Máquina y Aprendizaje Profundo.



**Figura 2.12:** Comparación de Inteligencia Artificial, Aprendizaje Máquina y Aprendizaje Profundo.

### 2.3.1. Aprendizaje Máquina

El Aprendizaje Máquina es la ciencia que busca construir programas de computadora que automáticamente mejoran con experiencia (Mitchell, 1997). La IA y ML son similares, pero no lo mismo; ya que, la IA es acerca de resolver problemas, razonamiento, y aprendizaje en general; por otro lado, ML es acerca de aprender a partir de ejemplos, definiciones y comportamientos (Kersting, 2018).

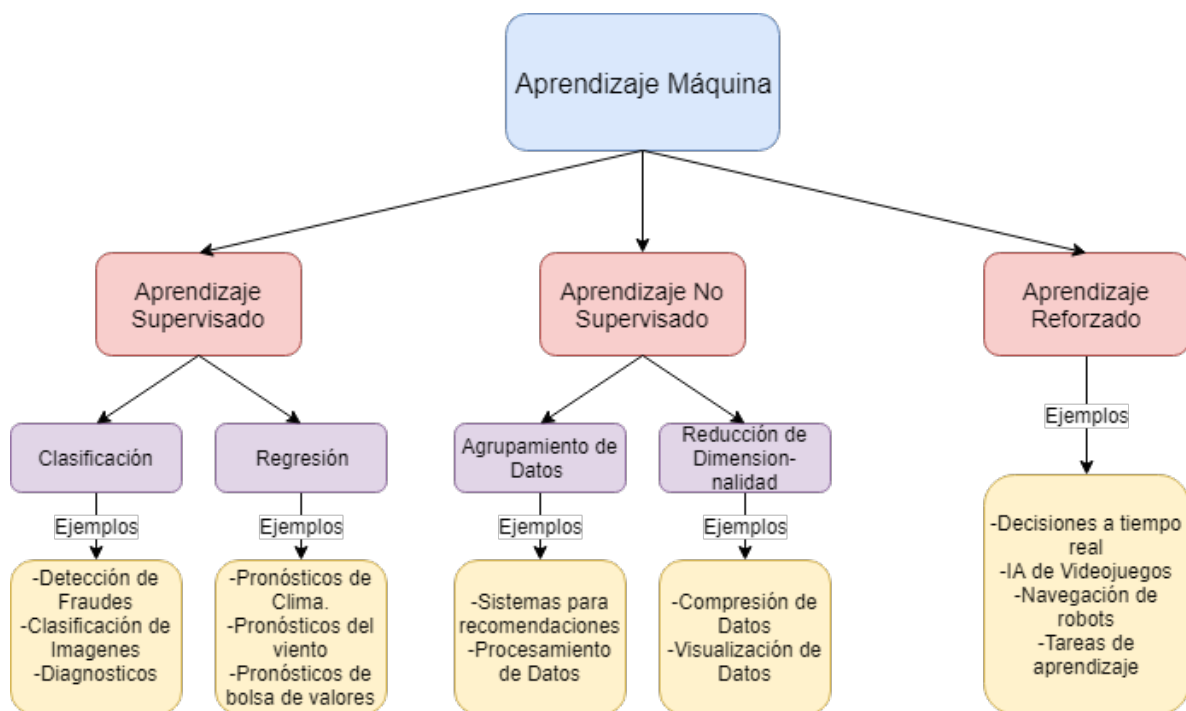
El uso de Aprendizaje Máquina se ha popularizado debido a la creciente cantidad de datos generados día a día, los cuales ayudan a mejorar los resultados obtenidos al utilizar ML para resolver problemas (Kersting, 2018).

El aprendizaje máquina se puede clasificar en:

- **Aprendizaje Supervisado:** Es la forma más común de Aprendizaje Profundo y Aprendizaje Máquina. Se caracteriza por entregar al modelo datos etiquetados, donde compara cada resultado obtenido contra los reales para corregir el error durante el entrenamiento (LeCun et al., 2015).

- **Aprendizaje No Supervisado:** Se caracteriza por no tener un etiquetado en sus datos que son utilizados para la corrección del modelo. Se utiliza para buscar extraer características de los datos de una distribución sin requerir esfuerzo humano (Goodfellow et al., 2016).
- **Aprendizaje Reforzado:** Involucra aprender que hacer, como mapeo de situaciones a acciones, para buscar maximizar la señal de recompensa (Sutton & Barto, 2018). Este método busca extrapolar o generalizar sus respuestas para actuar de manera correcta ante situaciones no presentes en datos de entrenamiento.

Para este trabajo se utiliza el Aprendizaje Supervisado (se utiliza k Vecinos más Cercanos), ya que las series de tiempo de viento se utilizan para entrenar el modelo para pronosticar el valor siguiente, es decir, se da al modelo el valor que requiere para ajustar el error. A pesar de ello, existe una gran cantidad de aplicaciones distintas para cada una de las diversas clasificaciones en el Aprendizaje Máquina, tales como IA en videojuegos con Aprendizaje reforzado (Oh et al., 2015), agrupamiento de datos con Aprendizaje No Supervisado (Jain, Dubes et al., 1988) y pronósticos de serie de tiempo con Aprendizaje Supervisado (Rangel



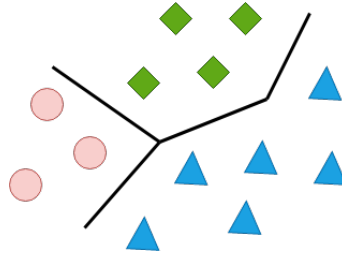
**Figura 2.13:** Tipos de Aprendizaje Máquina y ejemplos de aplicaciones.

et al., 2017).

En la Figura 2.13 se puede observar una gráfica de las ramificaciones para las clasificaciones de aprendizaje máquina y algunos ejemplos.

### 2.3.2. k Vecinos más cercanos

El algoritmo de k Vecinos más Cercanos es uno de los algoritmos de Aprendizaje Máquina más simples y básicos. Este algoritmo asigna la etiqueta a una observación  $x$  de su vecino más cercano (Figura 2.14). Por lo tanto, lo único que se debe hacer es calcular la distancia  $d(x, x')$  entre las observaciones (Smola & Vishwanathan, 2008).



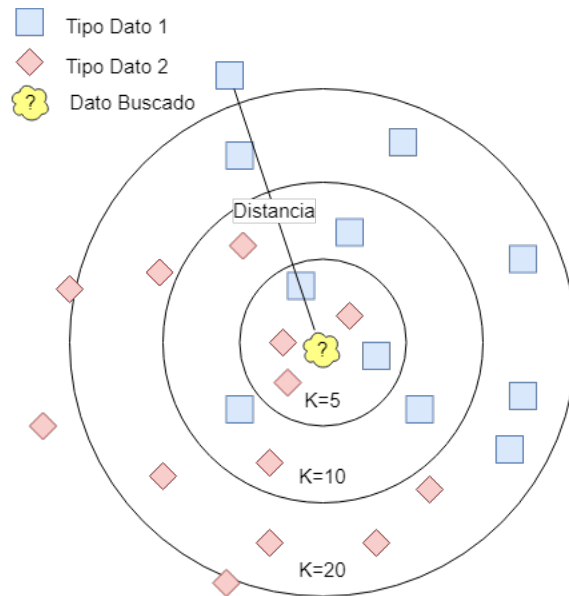
**Figura 2.14:** Representación básica de k Vecinos más Cercanos

Este algoritmo tiene como ventaja que las distancias no requieren ser simétricas, lo cual permite mayor flexibilidad; sin embargo, el problema con k vecinos más cercanos, son las estimaciones que pueden presentar ruido si una cantidad moderada de los datos utilizados son atípicos, por lo que son sensibles a la calidad de los datos (Smola & Vishwanathan, Smola & Vishwanathan).

Entre las distintas distancias más utilizadas que se pueden calcular para k Vecinos más Cercanos, se encuentra la distancia euclidiana para los  $k$  número de vecinos como se puede observar en la Figura 2.15.

### 2.3.3. Aprendizaje Profundo

El Aprendizaje Profundo es una forma de Aprendizaje Máquina que puede utilizar algoritmos de aprendizaje supervisado, no supervisado o ambos (LeCun et al., 2015). El Aprendizaje Profundo funciona de forma notable, ha impulsado de forma dramática el estado del arte en áreas desde reconocimiento del habla, traductores, reconocimiento de objetos, genómica,



**Figura 2.15:** Representación cálculo distancia para k Vecinos más Cercanos

entre otros (H. W. Lin et al., 2017). Las áreas en que se ha popularizado el uso de Aprendizaje Profundo han sido especialmente el procesamiento de lenguaje natural y visión computacional.

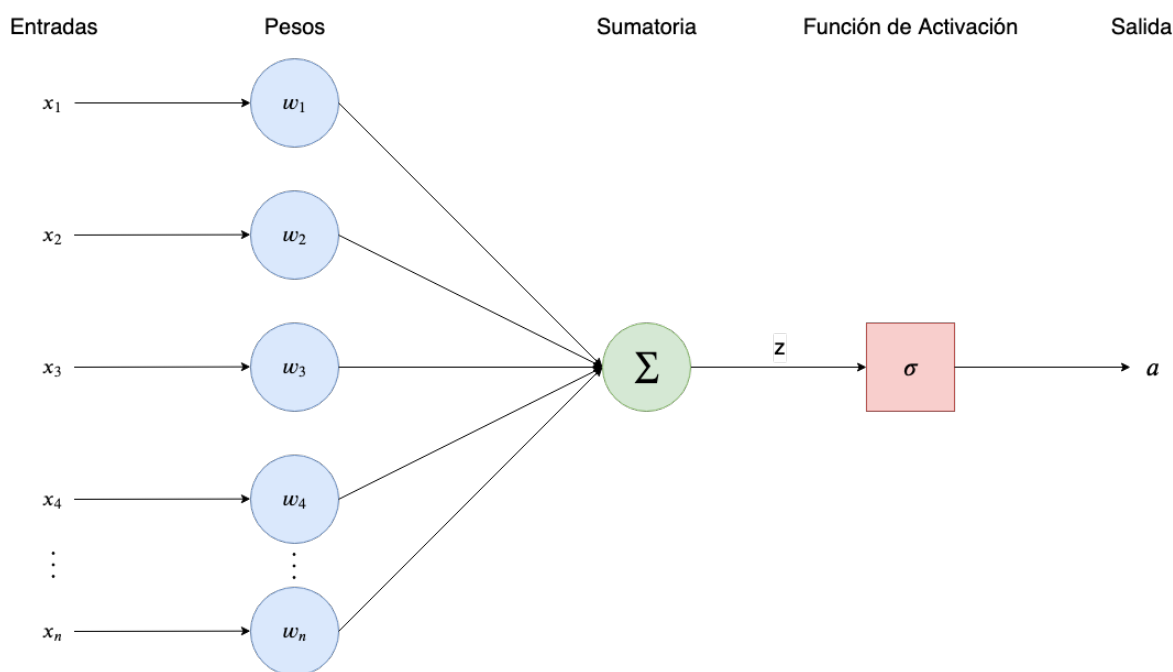
El Aprendizaje Profundo está basado en la rama del Aprendizaje Máquina que aprende por medio de extracción de características (LeCun et al., 2015). Por medio de la extracción de alto nivel y complejas abstracciones de representaciones de datos a través de un proceso jerárquico, el Aprendizaje Profundo obtiene resultados de forma rápida.

La palabra *profundo* de Aprendizaje Profundo (DL) proviene de los modelos de DL que son construidos con múltiples capas. Los modelos de DL debido a esta característica pueden tomar una gran cantidad de tiempo para entrenar. Sin embargo, el DL se popularizó debido a dos razones: el uso de tarjetas gráficas para computadoras o por su nombre en inglés *Graphic Processing Units* (GPU), que reducen el tiempo de entrenamiento requerido para los modelos y la segunda razón es debido a la gran cantidad de datos con los que se cuenta (LeCun et al., 2015). La gran cantidad de datos que se generan año con año crece, con esto aparecen retos que se pueden resolver utilizando aprendizaje profundo (X.-W. Chen & X. Lin, 2014).



### 2.3.3.1. Perceptrón Multicapas

El Perceptrón Multicapas (MLP) es una de las ANN más básicas, la cual está formada por múltiples Perceptrones (Figura 2.16). El Perceptrón básico fue desarrollado entre los años 1950s y 1960s por el científico Frank Rosenblatt, inspirado por trabajos tempranos de Warren McCulloch y Walter Pitts (Nielsen, 2015). El MLP es considerado el "hola mundo" por excelencia del DL (Goodfellow et al., 2016).

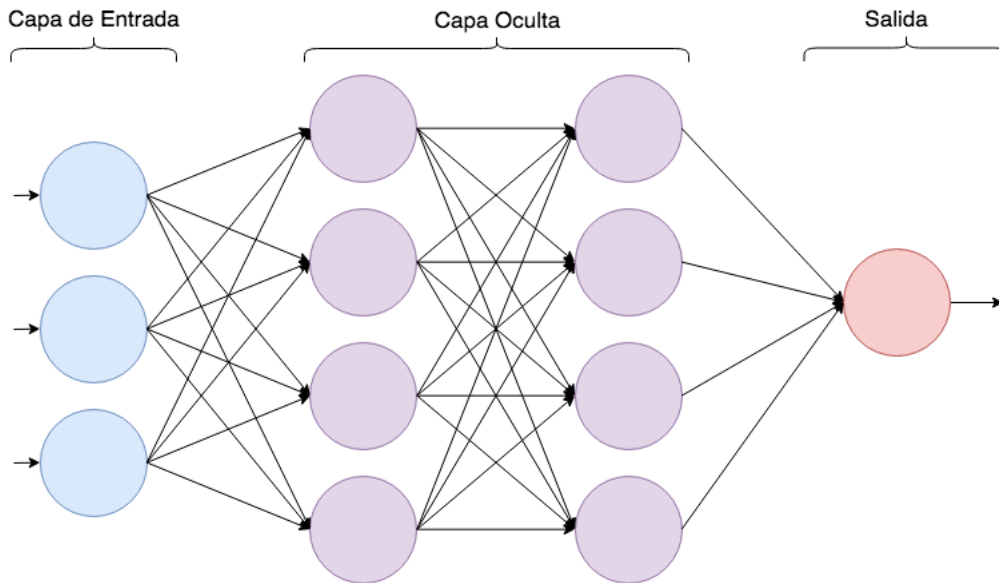


**Figura 2.16:** Estructura básica de un perceptrón

El funcionamiento de un Perceptrón es simple, toma múltiples entradas  $x_1, x_2, \dots, x_n$  y presenta una salida como se muestra en la Figura 2.16. La Ecuación matemática 2.12 representa el funcionamiento interno para el cálculo de la salida de un perceptrón, donde se aplica la función sigmoide a la sumatoria de  $w$  (pesos) que multiplican  $x$  entradas de datos.

$$salida = sigmoid\left(\sum_0^n w_i x_i\right) \quad (2.12)$$

La estructura de un MLP consiste de un mínimo de tres capas: entrada, oculta y salida (Figura 2.17). Los MLP son un tipo de red de alimentación hacia adelante o por su nombre en inglés *feedforward* que interconectan múltiples perceptrones.



**Figura 2.17:** Diagrama perceptrón multicapas básico para pronóstico de la velocidad del viento

Las MLP son un tipo de ANN capaces de modelar funciones complejas, son aptas para ignorar ruido y entradas irrelevantes, pueden adaptar sus pesos de forma sencilla además de ser fáciles de utilizar (Velo et al., 2014).

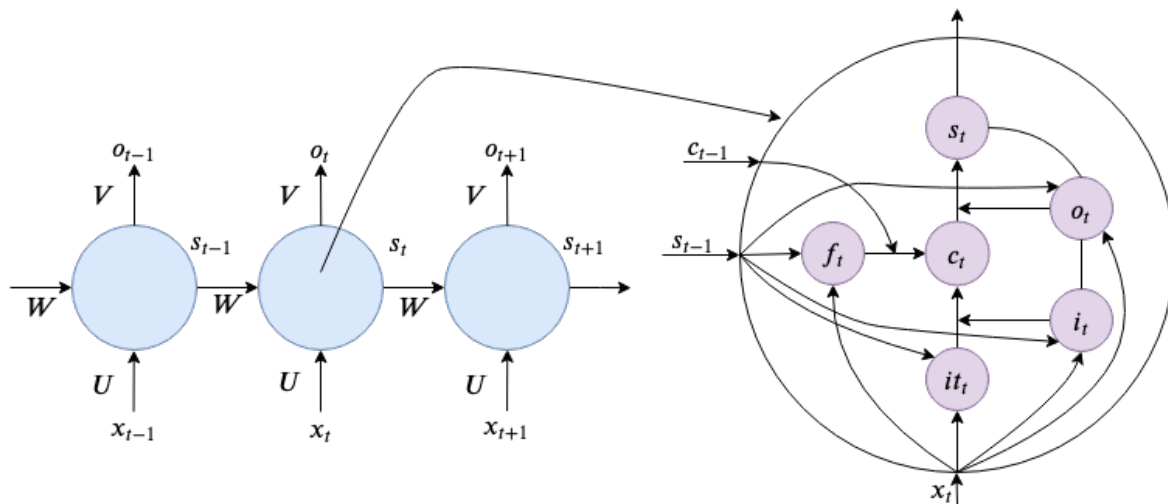
El proceso de aprendizaje de un MLP se puede dividir en dos fases principales, la alimentación de datos por medio de las entradas del MLP, y la corrección del error durante el cual se calculan los errores comparando los datos reales contra la respuesta que entrega el modelo por una técnica conocida como Propagación Hacia Atrás o por su nombre en inglés *Backpropagation*. Esta iteración se repite múltiples veces para reducir el error, utilizando un algoritmo para obtener un resultado con mejor aptitud, siendo la Regularización Bayesiana o *Bayesian Regularization* por su nombre en inglés, uno de los algoritmos más comunes (Velo et al., 2014).

### 2.3.3.2. Redes de gran memoria a corto plazo

Las redes de gran memoria a corto plazo o por su nombre en inglés *Long Short-Term Memory* (LSTM) son muy similares a las MLP en su estructura, tienen capas de entrada, capas ocultas y capa de salida. Sin embargo, LSTM en su capa oculta poseen una unidad de memoria (Xiaoyun et al., 2016).

Las LSTM son un tipo ANN que se clasifican como Redes Neuronales Recurrentes, que se caracterizan por no ser estrictamente *feed forward* tal como las MLP. Esto debido a que las LSTM utilizan entradas de iteraciones anteriores para futuros cálculos de salidas realizando así una retroalimentación. Este tipo de ANN son potencialmente más poderosas que las MLP y se caracterizan por mostrar un comportamiento temporal (Hagan et al., 1996).

La unidad de memoria consiste de tres compuertas: compuerta de entrada, compuerta de olvido y compuerta de salida (*Input Gate* ( $i_t$ ), *Forget Gate* ( $f_t$ ) y *Output Gate* ( $o_t$ )) y una conexión recurrente (Figura 2.18). La unidad tiene una entrada  $x_t$  y dos retroalimentaciones de estados previos de la unidad que son la salida del estado previo  $s_{t-1}$  y la variable de estado  $c_{t-1}$ . Las compuertas utilizan una función de activación sigmoide  $g$ , mientras que los estados utilizan una función  $\tanh$ . La unidad de memoria de una LSTM se puede definir por el conjunto de Ecuaciones 2.13, donde  $W$  son parámetros y  $b$  es el sesgó (Xiaoyun et al., 2016).



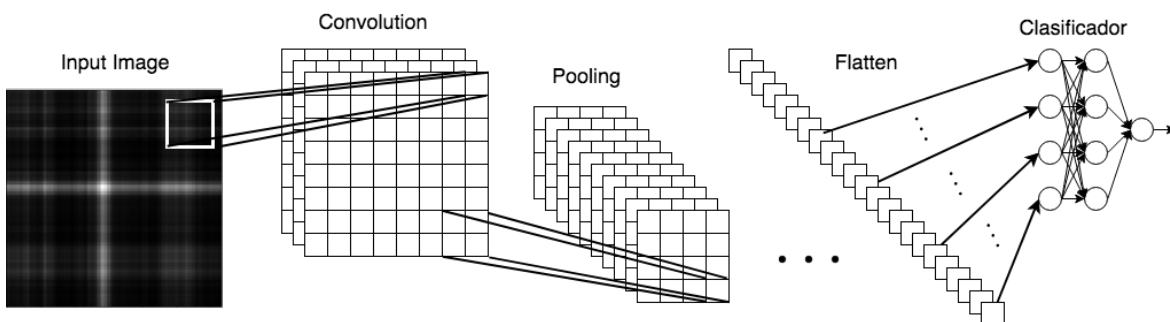
**Figura 2.18:** Diagrama de funcionamiento de una celda de las Redes de Gran Memoria a Corto Plazo

$$\begin{aligned}
\text{Input Gate} & \quad i_t = g(W_{xi}x_t + W_{hi}s_{t-1} + b_i) \\
\text{Forget Gate} & \quad f_t = g(W_{xf}x_t + W_{hf}s_{t-1} + b_f) \\
\text{Output Gate} & \quad o_t = g(W_{xo}x_t + W_{ho}s_{t-1} + b_o) \\
\text{Input Transform} & \quad it_t = \tanh(W_{xc}x_t + W_{hc}s_{t-1} + b_{it_t}) \\
\text{State Update} & \quad c_t = f_t \cdot c_{t-1} + i_t \cdot it_t \\
\text{Output} & \quad s_t = o_t \cdot \tanh(c_t)
\end{aligned}
\tag{2.13}$$

### 2.3.3.3. Redes Neuronales Convolucionadas

Las Redes Neuronales Convolucionadas o por su nombre en inglés *Convolutional Neural Networks* (CNN) son capaces de extraer características de imágenes de dos dimensiones (Zhu et al., 2017). Las CNN son un tipo de ANN que se especializan en procesar datos con una topología de tipo cuadrangular (dos dimensiones). Estas redes son simplemente redes neuronales que utilizan la Convolución en lugar de la multiplicación general de matrices que es común en otras ANN en alguna de sus capas; donde, la Convolución es un tipo de operación lineal. Las CNN se conforman de cuatro etapas (Figura 2.19) principales:

1. Convolución
2. *Pooling*
3. *Flatten*
4. Clasificador

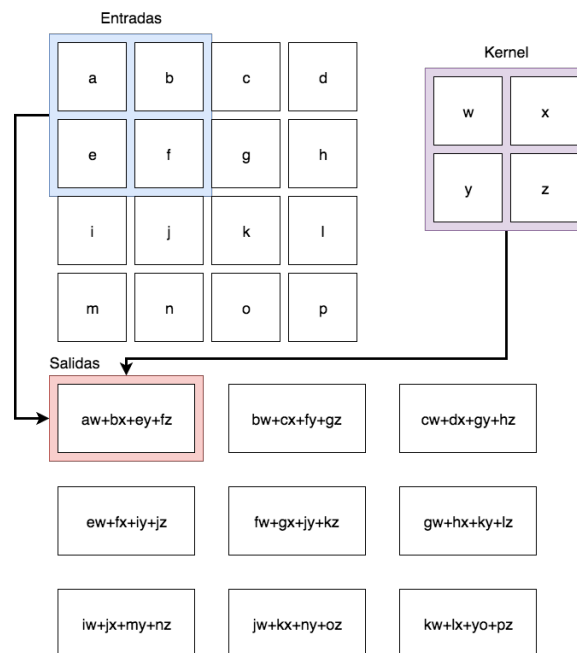


**Figura 2.19:** Diagrama de funcionamiento de una Red Neuronal Convolucionada

La **Convolución** es el primer paso que realizan las CNN, este paso consiste en una operación de dos funciones de un argumento de tiempo real (Goodfellow et al., 2016). La Convolución se puede definir matemáticamente por la Ecuación 2.14, donde el primer término  $x$  se conoce como entradas, el segundo término  $w$  como *kernel* y la salida  $s$  mapa de características o por su nombre en inglés *feature map*.

$$s(t) = (x * w)(t) \quad (2.14)$$

En aplicaciones de ML la entrada de datos es usualmente un arreglo de datos multidimensional y el *kernel* un arreglo multidimensional de parámetros que son adaptados por el algoritmo de aprendizaje (Goodfellow et al., 2016). El proceso de Convolución consiste en realizar una cantidad de multiplicaciones de estos arreglos multidimensionales, por lo que la carga computacional se encuentra principalmente en la capa de convolución. En la Figura 2.20 se muestra un ejemplo del proceso de Convolución, donde se realiza una operación de multiplicación del kernel por las entradas en la posición actual, posteriormente realizándose una suma de estas multiplicaciones para generar un nuevo mapa de características.

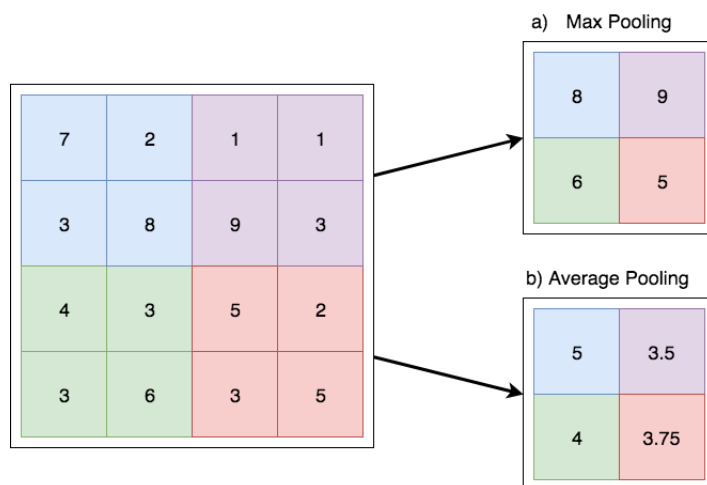


**Figura 2.20:** Ejemplo de proceso de convolución, donde se muestra multiplicación de entradas con kernel para la obtención de la salida de mapa de características.

La función **Pooling** reemplaza la salida de la red con un resumen estadístico de las salidas

cercanas (Goodfellow et al., 2016). Este proceso tiene como objetivo convertir la representación de los datos de salida de la etapa de convolución a una representación invariante más pequeña, esto con el objetivo de poder trabajar con entradas de tamaño variable, reducir la dimensión de las mismas y reducir el poder computacional requerido. Existen dos tipos de pooling (Figura 2.21):

- **Max Pooling:** Toma el valor máximo entre los valores que toma como muestra (Figura 2.21 a).
- **Average Pooling:** Calcula la media de los valores tomados como muestra (Figura 2.21 b).



**Figura 2.21:** Ejemplo de los tipos de pooling: a) Max Pooling y b) Average Pooling

La etapa de *Flatten* es un proceso donde se toman todas las colecciones de datos, se extraen sus datos y todos estos datos son almacenados en una sola colección. Finalmente, la etapa del clasificador es el simple uso de algún modelo de aprendizaje, el cual utiliza las características extraídas durante el proceso de convolución.

## 2.4. Optimización de hiperparámetros

El uso de técnicas de IA tales como aprendizaje profundo (CNN, MLP y LSTM) o técnicas de aprendizaje máquina han permitido la obtención de excelentes resultados en problemas

de clasificación y regresión, esto gracias a su ajuste automático de parámetros internos por medio de algoritmos de aprendizaje supervisados (Young et al., 2015). Sin embargo, la tarea de ajustar los hiperparámetros, los cuales son los que definen la arquitectura del modelo, tamaño de datos con los que trabaja, topología de la red, entre otros, es tediosa y consume grandes cantidades de tiempo. Es por esto que se busca utilizar técnicas que permitan automatizar este proceso. Para este trabajo se utilizan las técnicas de Algoritmos Genéticos (AG) y Búsqueda Aleatoria o por su nombre en inglés *Random Search* (RS).

### 2.4.1. Algoritmos Genéticos

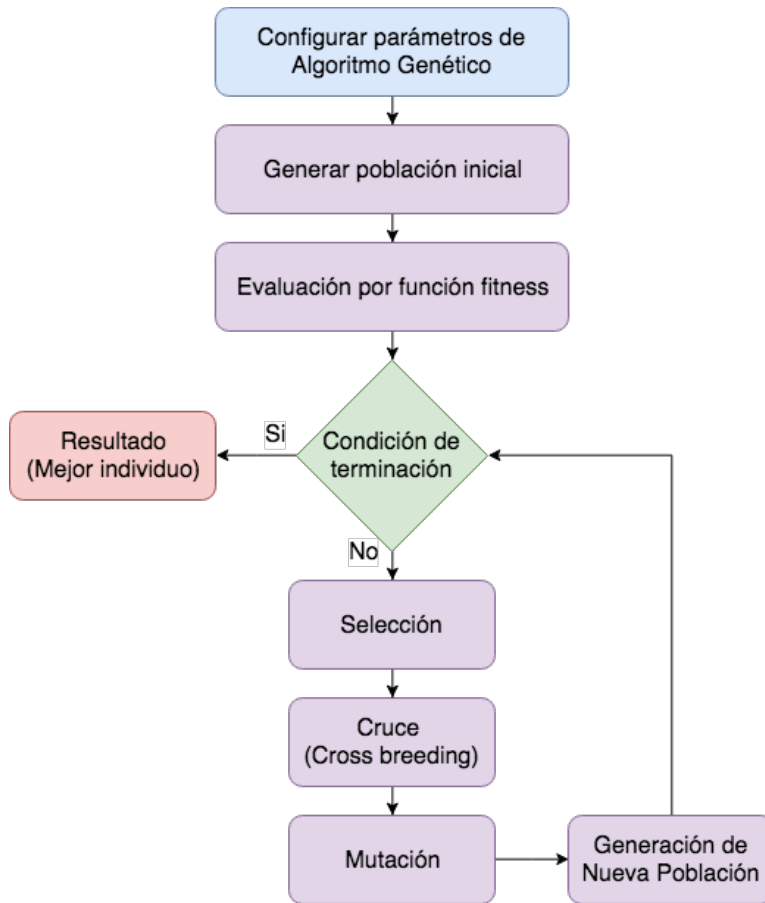
Los Algoritmos Genéticos (AG) son métodos adaptativos que pueden ser utilizados para la solución de problemas de optimización y búsqueda (Beasley et al., 1993). Los AG están basados en el proceso genético de los organismos biológicos; ya que utilizan una analogía directa su comportamiento natural. Estos trabajan con poblaciones de **individuos**, los cuales representan una solución posible a un problema dado y cada uno de ellos les es dada una calificación para su nivel de aptitud (o *fitness* por su nombre en inglés).

El proceso general de un AG consiste de 6 pasos principales:

- Generación de población inicial
- Evaluación de aptitud por *función fitness*
- Selección de mejores individuos
- Cruce (Cross breeding)
- Mutación
- Generación de nueva población

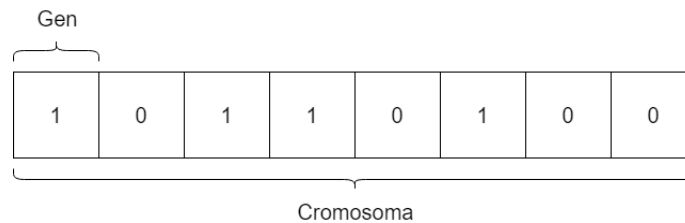
Los parámetros de un AG deben configurarse previo al inicio del proceso y debe programarse una condición de terminación. En la Figura 2.22 se puede observar el proceso completo que sigue un AG básico.

Un individuo es una representación para la solución de un problema, el cual esta compuesto por múltiples cromosomas. Cada uno de estos cromosomas representa una parte de



**Figura 2.22:** Diagrama de proceso básico para un algoritmo genético

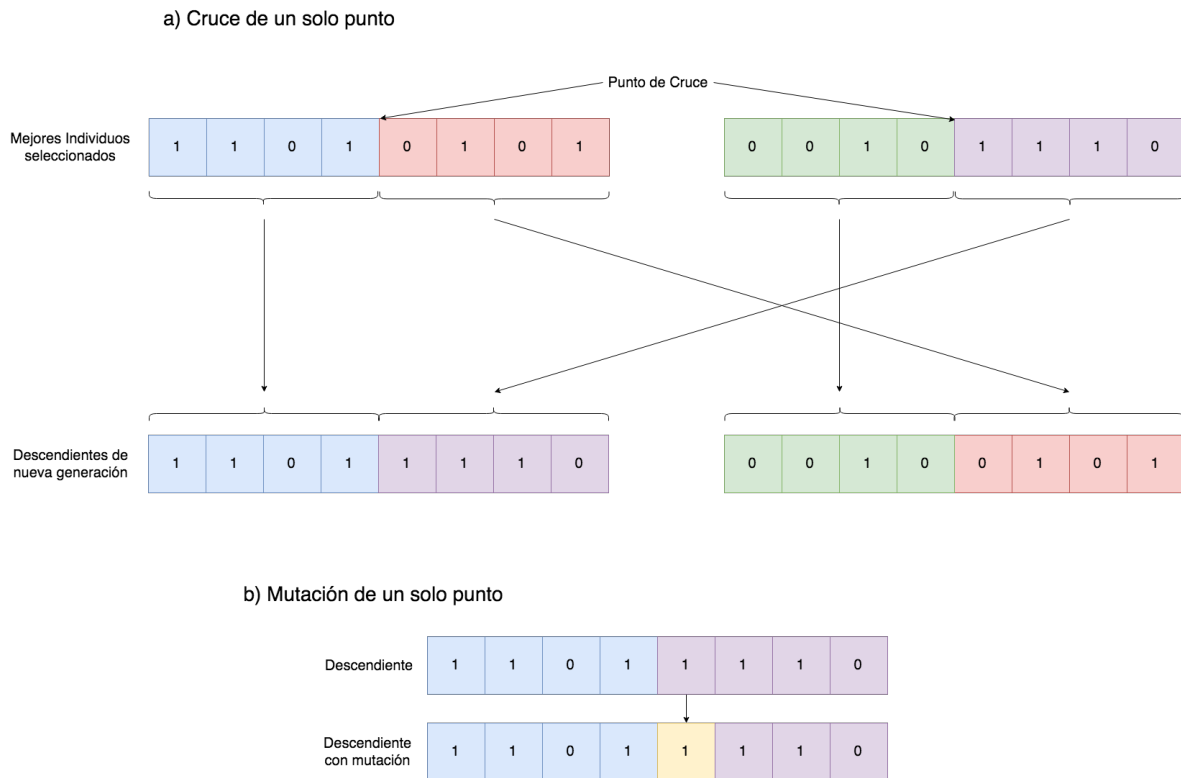
la solución del problema. Un cromosoma se puede conformar de múltiples **genes** que utilizan distintos valores de acuerdo al problema que se intenta solucionar, tales como números binarios o decimales (Figura 2.23).



**Figura 2.23:** Ejemplo simple de un individuo con genes binarios

El proceso para calificar el nivel de aptitud para un individuo se realiza por medio de la *función fitness*, la cual regresa un valor numérico que califica el individuo que es proporcional a la utilidad o habilidad de este individuo. Para cada problema en específico a resolver debe diseñarse una *función fitness* específica.





**Figura 2.24:** Ejemplos de cruce de individuos (a) y mutación de individuo (b)

Los individuos con alto nivel de aptitud son utilizados en un proceso de recombinación, generando así una nueva población de individuos **descendientes**. Los operadores básicos para la generación de una nueva población son:

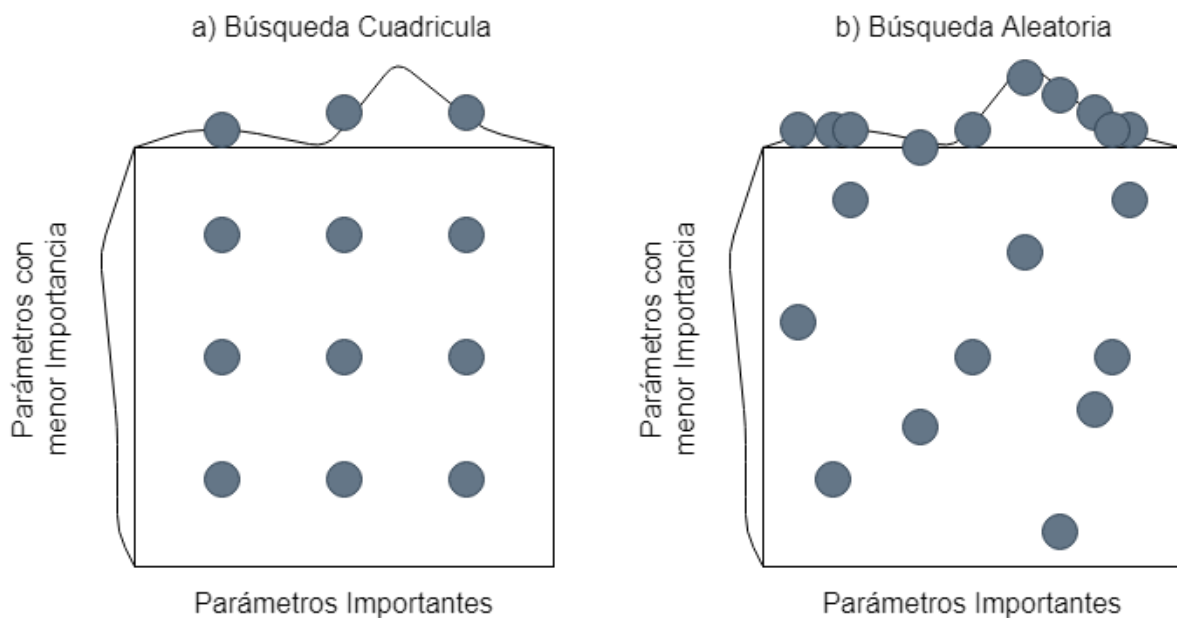
- **Cruce** (Cross over): Toma dos individuos, corta sus cromosomas en un punto aleatorio para producir dos *cabezas* y *colas*. Posteriormente procede a intercambiar estas cabezas y colas de los individuos para generar nuevos descendientes (Figura 2.24 a).
- **Mutación**: Se aplica a cada descendiente de forma individual. Altera cada gen del cromosoma de forma aleatoria (Figura 2.24 b).

El AG es correctamente implementado si se acerca a un punto de **convergencia** a través de la evolución y sucesión de múltiples generaciones, mejorando su aptitud y acercándose cada vez más a un punto global óptimo. Se dice que un gen (en este trabajo siendo un gen equivalente a un hiperparámetro) converge cuando el 95 % de la población comparte el mismo valor, y una población se dice que converge cuando todos los genes han convergido (Young et al., 2015).

## 2.4.2. Optimización de hiperparámetros por caja negra

En general cualquier método de optimización de hiperparámetros de caja negra o por su nombre en inglés *Blackbox Parameter Optimization* (BPO) puede ser utilizado para la solución de optimización global de hiperparámetros (Feurer & Hutter, 2019). Para este trabajo se utilizan los métodos básicos de BPO, los cuales son:

- **Búsqueda de Cuadrícula** o por su nombre en inglés *Grid Search* (GS): Para este método el usuario define una cantidad finita de valores para cada hiperparámetro que se busca, posteriormente GS evalúa el producto cartesiano de estos conjuntos (Figura 2.25 a). Este método sufre de problema de dimensionalidad, esto debido a que el incremento de la cantidad de parámetros que se deben evaluar incrementa el poder computacional requerido proporcionalmente.
- **Búsqueda Aleatoria** o por su nombre en inglés *Random Search* (RS): Este método toma configuraciones finitas similarmente a GS; sin embargo, el RS toma puntos aleatorios de esta red para realizar evaluaciones (Figura 2.25 b). El método de RS funciona de mejor manera que GS cuando algunos hiperparámetros son más importantes que



**Figura 2.25:** Importancia de hiperparámetros para Búsqueda de Cuadrícula (GS) y Búsqueda Aleatoria (RS)

otros (Feurer & Hutter 2019).

El uso de GS y RS se consideran métodos de búsqueda por fuerza bruta, esto debido a que solamente siguen un orden sistemático (GS) o buscan de forma aleatoria (RS), por lo que se consideran técnicas que rara vez se utilizan por si solas (Young et al., 2015).

## 2.5. Métricas estadísticas

El cálculo del error para un modelo pronosticador es importante ya que ayuda a tener un criterio durante la evaluación de dichos proyectos. Múltiples criterios para el cálculo del error han sido utilizados para calificar la aptitud de los modelos (Hyndman & Koehler, 2006). Un conjunto tradicional para la medición de error es utilizado en múltiples disciplinas a pesar de sus inconvenientes debido a la necesidad de verificar la utilidad que presenta un modelo pronosticador para su uso práctico (Shcherbakov et al., 2013).

Las ecuaciones utilizadas en este trabajo para calcular el error de los modelos pronosticadores son:

- Error Cuadrático Medio o por su nombre en inglés *Mean Squared Error* (MSE).
- Error Absoluto Medio o por su nombre en inglés *Mean Absolute Error* (MAE).
- Raíz Error Cuadrático Medio o por su nombre en inglés *Root Mean Squared Error*.

Estas ecuaciones se basan en el cálculo del error absoluto, donde incluye el cálculo el estimado del valor  $e_i$  dada por la Ecuación 2.15.

$$e_t = (y_t - \hat{y}_t) \quad (2.15)$$

donde  $e_t$  es el error calculado de la diferencia entre el valor real ( $y_t$ ) y el valor predicho ( $\hat{f}_t$ ) medido en el tiempo  $t$ . (Shcherbakov et al., 2013).

El calculo del error se da por medio del conjunto de Ecuaciones 2.16.

$$\begin{aligned} a) \text{MSE} &= \frac{1}{n} \sum_{i=1}^n e_i^2 \\ b) \text{MAE} &= \frac{1}{n} \sum_{i=1}^n |e_i| \\ c) \text{RMSE} &= \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2} \end{aligned} \tag{2.16}$$

---

# Capítulo 3

## Estado del Arte

---

Este capítulo recopila literatura estudiada de distintos métodos de pronóstico a corto plazo y métodos utilizados para la optimización de hiperparámetros para métodos de IA. Los trabajos que fueron consultados tratan los distintos métodos de pronóstico existentes, tanto modelos físicos, estadísticos e híbridos; además, fueron consultados trabajos relacionados a la optimización de hiperparámetros utilizados en modelos de IA.

### 3.1. Métodos de pronóstico

El uso de pronósticos de la velocidad y poder generado por el viento tiene muchos años de evolución, debido a la necesidad de mejorar el proceso de generación de energía por medios eólicos (Wu & Hong, 2007). Dada la necesidad de producir este tipo de energía de forma confiable, económica, y estable, tal que permita reemplazar el uso de medios convencionales, como el uso de recursos fósiles utilizados en la producción de energía (Lei et al., 2009). El uso de técnicas de pronóstico ha demostrado llevar hasta un 10 % de ahorro el costo de producción energética en granjas eólicas (Soman et al., 2010).

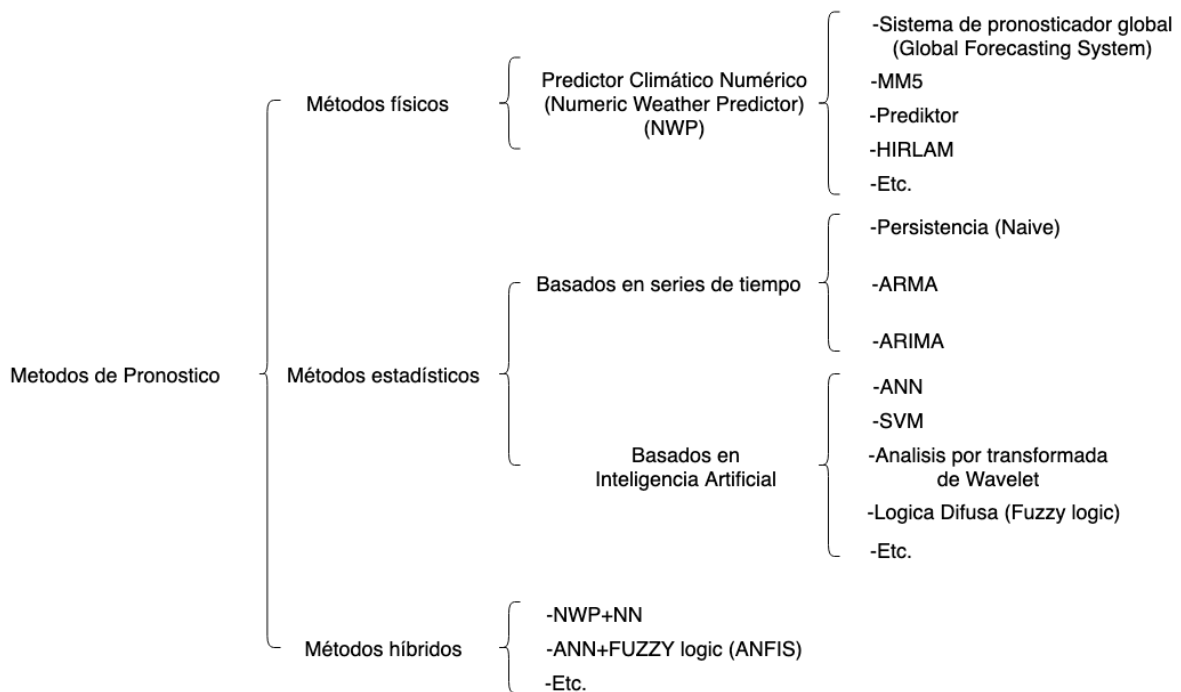
Mao & Shaoshuai, en 2016 muestran que los métodos de pronóstico se pueden clasificar de acuerdo a distintos criterios, tales como horizonte pronosticado, tipo de modelo pronosticador, precisión de salidas, tipos de dato de entrada y rango de terreno en el que se realiza un pronóstico efectivo. En la Tabla 3.1 se muestran las diferentes clasificaciones y algunos ejemplos para cada uno de estos.

En sus revisiones del estado del arte Soman et al. (2010) y Mao & Shaoshuai, (2016)

**Tabla 3.1:** Distintas clasificaciones por distintos criterios para métodos de pronóstico de viento.

Tipo de Clasificación	Ejemplos de Clasificación
Clasificación por escala de tiempo	Ultra corto plazo
	Corto plazo
	Medio plazo
	Largo plazo
Clasificación por modelo pronosticador	Métodos físicos
	Métodos estadísticos
	Métodos híbridos
Clasificación por precisión de salidas	Pronósticos determinísticos
	Pronósticos con incertidumbre
Clasificación por tipo de dato de entrada	Simulación numérica
	Datos históricos
Clasificación por rango de pronóstico	pronóstico de una sola ubicación
	pronóstico de una granja eólica completa
	pronóstico de una región completa

clasifican a los métodos para pronóstico en tres grandes grupos: métodos físicos, métodos estadísticos y métodos híbridos (Figura 3.1). Los métodos físicos se caracterizan por utilizar múltiples datos numéricos tales como temperatura, presión atmosférica, superficie y obstáculos, los métodos estadísticos se basan en el entrenamiento de modelos con datos históricos para afinarlos y utilizar datos actuales para realizar un pronóstico, finalmente los métodos



**Figura 3.1:** Diagrama de árbol de clasificación de modelos de pronóstico (datos tomados de (Soman et al. 2010) y (Mao & Shaoshuai 2016))

híbridos buscan combinar la fortaleza de distintos acercamientos para obtener resultados con mejor aptitud (Soman et al., 2010).

Los métodos físicos se caracterizan por utilizar predictores climáticos numéricos ó por su nombre en inglés *Numeric Weather Prediction*(NWP). Estos métodos se caracterizan por obtener muy buenos pronósticos cuando las condiciones climáticas del área de interés son estables (Soman et al., 2010). Los métodos que utilizan NWP tienen el defecto de requerir el uso de súper computadores debido a las complicadas operaciones matemáticas que son requeridas para su uso (Potter & Negnevitsky 2006). Sin embargo, los métodos físicos son utilizados por grandes compañías que ofrecen servicios de estudio y pronósticos para el viento tal como MetoTemp que es una rama de MeteoConsult, la cual opera en los países bajos de Europa (Bhaskar et al., 2010).

Los métodos estadísticos se dividen en dos grandes grupos: basados en series de tiempo y basados en inteligencia artificial. Estos métodos se caracterizan por ser fáciles de modelar, no son costosos y proveen pronósticos oportunos (Soman et al., 2010).

Entre los modelos basados en series de tiempo destacan: Auto Regresivo (AR), Auto Regresivo Media Móvil (ARMA), Auto Regresivo con Media Móvil Integrada (ARIMA), Naive, acercamiento bayesiano y pronósticos grises (Chang, 2014).

Erdem & Shi, en 2011 muestra cuatro acercamientos distintos para el pronóstico de la velocidad y dirección del viento, donde la métrica utilizada para calcular el error fue MAE. Los modelos utilizados fueron basados en ARMA y autoregresión por vector (VAR), donde se utilizó una variación para cada uno de estos. Los resultados mostraron mejores resultados utilizando VAR.

El modelo ARIMA es utilizado por Kavasseri & Seetharaman, en 2009 para el pronóstico de 24h y 48h (uno y dos días). El modelo de ARIMA utilizado es un modelo funcional  $f$ -ARIMA con datos históricos de la velocidad del viento en Dakota del Norte. Los resultados fueron comparados con el método de pronóstico de persistencia (Naive), los cuales mostraron tener resultados con mejor aptitud.

Los modelos basados en IA han demostrado gran flexibilidad y capacidad para los pronósticos a corto plazo. Entre los modelos de IA los que mas destacan son los modelos de DL los cuales han crecido en popularidad gracias al avance tecnológico y uso de paralelización con

GPU (unidad de procesamiento gráfico) para el entrenamiento de los modelos pronosticadores (LeCun et al., 2015).

En el trabajo de Flores et al., 2012, se utiliza un acercamiento evolutivo en conjunto con modelos ANN y ARMA. El proceso realizado fue automatizado gracias al acercamiento evolutivo, y los resultados muestran que los modelos ANN fueron en todos los casos los que mostraron mejor aptitud.

Santamaría-Bonfil et al., en 2016 utilizó un acercamiento híbrido con SVM utilizando datos históricos dados por *Mexican Wind Energy Technology Center* (CERTE) de la localidad de La Ventosa para el entrenamiento los modelos, comparan sus resultados con técnicas de AR, ARMA, ARIMA y Naive. Los resultados muestran que los pronósticos obtenidos por su propuesta muestran mejor aptitud cuando se realizan pronósticos de 5 a 23 horas.

En su trabajo Dalto et al., en 2015 muestra técnicas de DL y ANN simples para realizar pronósticos de la velocidad del viento. Los resultados muestran que las ANN simples sufren pérdida de precisión debido a variables de entrada redundantes e irrelevantes, en comparación de las implementaciones que utilizan DL.

A diferencia de pronosticar el paso siguiente en el tiempo, los pronósticos con un horizonte mas amplio son mas difíciles porque el error y la incertidumbre se acumulan, lo que reduce la precisión (Taieb, Bontempi, Atiya et al., 2012). Además, la estrategia que use un pronosticador es de vital importancia, ya que afecta el costo computacional y su precisión.

Pasero et al., en 2010 implementa un enfoque de pronóstico a largo plazo con un modelo MLP y una estrategia recursiva; sin embargo, sus resultados muestran que un modelo de SVM con estrategia directa consigue mejores resultados. Por otro lado, las LSTM tienen la característica de poder procesar grandes cantidades de datos y poseen la capacidad de generalización; es decir, se adaptan a datos no conocidos. Estos obtienen mejores resultados que los modelos basados en SVM (Xiaoyun et al., 2016).

Zhu et al., en 2017 utiliza las CNN para realizar pronósticos de viento, donde la entrada de datos utilizados se acomodan en matrices de dos dimensiones para realizar el entrenamiento del modelo pronosticador. El experimento realizado fue para pronósticos de 4 horas los cuales muestran una aptitud aceptable.

En el trabajo de Bottieau et al., en 2018 utiliza LSTM para realizar pronósticos de la



velocidad del viento en granjas eólicas de 24 horas. Este trabajo compara los resultados obtenidos por las LSTM con MLP, versiones profundas de MLP y métodos clásicos. Las LSTM muestran obtener pronósticos con mejor aptitud que las MLP y métodos clásicos.

Otra método que ha crecido en popularidad en años recientes ha sido el uso de la transformada de Wavelet en conjunto con otros métodos para realizar pronósticos de la velocidad del viento. En su trabajo H. Liu, Tian, Pan et al., en 2013 utiliza la transformada de Wavelet junto a ANN, donde compara sus resultados con modelos clásicos de ANFIS, ARMA y persistencias. Los resultados mostrados por su propuesta muestran aptitud satisfactoria comparados con los modelos clásicos.

En su trabajo Soman et al., en 2010 muestra la comparación de distintos acercamientos utilizados para realizar pronósticos de la velocidad del viento. En la Tabla 3.2 se muestran los distintos acercamientos y sus subclases; además, se muestran algunos ejemplos y observaciones para los acercamientos presentados en su trabajo.

**Tabla 3.2:** Tabla de comparación de acercamientos para realizar pronósticos de la velocidad del viento descrita por Soman et al., 2010.

Método de pronóstico	Subclase	Ejemplos	Observaciones
Método de persistencia	-	$P(t+k) = P(t)$	●Preciso para modelos de ultra corto plazo.
Acercamientos físicos	Predictores de clima numéricos o por su nombre en inglés <i>Numeric Weather Predictors</i> (NWP)	<ul style="list-style-type: none"> <li>●Sistema de pronóstico global</li> <li>●MM5</li> <li>●Prediktor</li> <li>●HIRLAM</li> <li>●etc.</li> </ul>	●Utilizan datos meteorológicos tales como velocidad y dirección del viento, presión, temperatura, humedad, estructura del terreno, etc.
Acercamientos estadísticos	Redes Neuronales Artificiales(ANN)	<ul style="list-style-type: none"> <li>●MLP</li> <li>●Recurrentes</li> <li>●CNN</li> <li>●ADALINE</li> <li>●etc.</li> </ul>	<ul style="list-style-type: none"> <li>●Precisos para modelos de corto plazo.</li> <li>●Usualmente superan a los modelos basados en series de tiempo.</li> </ul>
	Modelos basados en series de tiempo	<ul style="list-style-type: none"> <li>●ARMA</li> <li>●ARIMA</li> </ul>	<ul style="list-style-type: none"> <li>●Precisos para corto plazo</li> <li>●En algunas situaciones superan a las ANN</li> </ul>
Algunas nuevas técnicas	-	<ul style="list-style-type: none"> <li>●Correlación espacial</li> <li>●Lógica difusa</li> <li>●Transformada de Wavelet</li> <li>●Predicciones de conjunto</li> <li>●Predicciones de conjunto</li> <li>●Entrenamiento basado en entropía</li> </ul>	<ul style="list-style-type: none"> <li>●Entrenamiento de modelos basado en entropía mejora los resultados obtenidos por los modelos</li> <li>●Utilizan técnicas mas novedosas.</li> </ul>
Acercamientos híbridos	-	<ul style="list-style-type: none"> <li>●NWP+ANN</li> <li>●ANFIS</li> <li>●NWP+series de tiempo</li> </ul>	●Combinan fortalezas de distintos modelos

## 3.2. Métodos de optimización de hiperparámetros

La mayoría de los métodos de pronóstico que utilizan IA tienen hiperparámetros que deben ser ajustados para obtener resultados deseables; debido a esto, se utilizan técnicas para automatizar el proceso de ajuste de estos hiperparámetros para optimizar y obtener un mejor rendimiento (Feurer & Hutter, 2019).

Young et al., en 2015 utilizó algoritmos genéticos para realizar ajustes automáticos de hiperparámetros a modelos de DL. Para esto designa los distintos hiperparámetros con los que cuenta el modelo y asigna cada uno de estos a un gen específico. Genera una población inicial, y a través de las generaciones por las cuales se van seleccionando los mejores individuos se obtienen hiperparámetros optimizados para este modelo. Esto ayuda a eliminar la tediosa tarea que es configurar los hiperparámetros de forma manual, el cual es un proceso que puede tomar varias horas.

En su trabajo D. Liu et al., en 2014 utiliza algoritmos genéticos para optimizar los hiperparámetros de sus modelos que utilizan la transformada de Wavelet. Los resultados obtenidos por su implementación y optimizados por AG obtienen mejores resultados que aquellos de los métodos clásicos y que no utilizan optimización por AG.

Saroha & Aggarwal, en 2014 presenta una comparación de tres distintos modelos de ANN para pronósticos de múltiples pasos en el tiempo. El algoritmo genético utiliza una evaluación fitness utilizando la medida de calidad MAE. Los resultados mostraron tener mejor aptitud que los resultados obtenidos por métodos clásicos.

A pesar de la ventaja que brinda el uso de técnicas para la optimización de hiperparámetros, estos aun tienen el defecto de requerir grandes cantidades tiempo requerido para entrenar un modelo de acuerdo a la cantidad de hiperparámetros que se desean ajustar (Feurer & Hutter, 2019). Es por esto que trabajos como el de Salza et al., en 2016 proponen el uso de estrategias de paralelización para algoritmos de optimización de hiperparámetros. El uso de la paralelización pueden disminuir de gran manera la cantidad de tiempo requerido para entrenar un solo modelo, lo que puede ayudar a reducir los costos que conlleva el optimizar modelos pronosticadores.

---

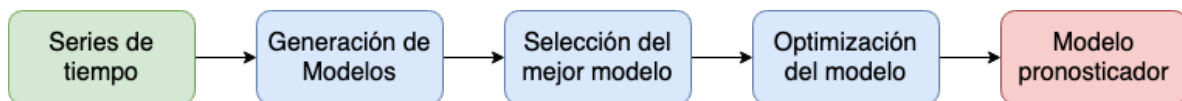
# Capítulo 4

## Metodología Propuesta

---

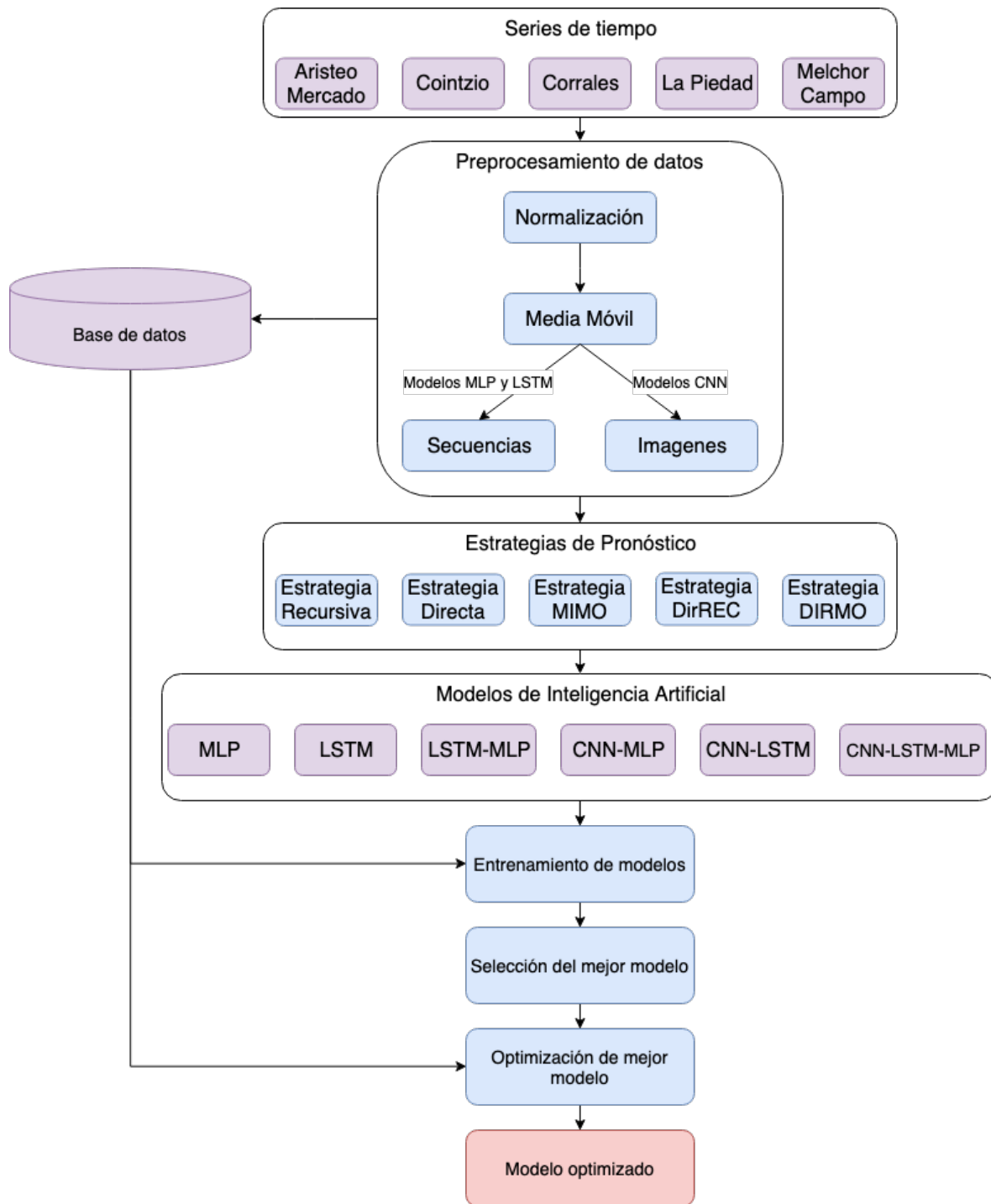
El Capítulo 4 describe como se diseñó la metodología para pronósticos de viento de uno a múltiples pasos en el tiempo ( $1 \leq t \leq 24$ ) utilizando modelos de Inteligencia Artificial. Así mismo, se describen las estrategias utilizadas para modelos de predicción, preprocesamiento de datos, técnicas de optimización de hiperparámetros y tecnologías utilizadas en este trabajo para mejorar la aptitud del resultado obtenido.

En la Figura 4.1 se muestran los pasos realizados por la metodología para obtener el modelo pronosticador. El proceso consiste a partir de datos históricos (series de tiempo) de la velocidad del viento se generan múltiples modelos combinando las distintas estrategias de pronóstico y modelos de Inteligencia Artificial, posteriormente estos son comparados utilizando la métrica de MSE. Finalmente, el modelo seleccionado es optimizado utilizando Algoritmos Genéticos (AG) inicializado por búsqueda aleatoria (RS) para obtener el modelo objetivo.



**Figura 4.1:** Pasos seguidos por la metodología propuesta

La Figura 4.2 muestra el proceso utilizado en este trabajo para generar un modelo pronosticador para la series de tiempo de la velocidad del viento, iniciando con las series de tiempo y concluyendo con la obtención de un modelo de pronosticador para  $n$  pasos futuros. El proceso consiste cuatro etapas principales; preprocesamiento de datos, selección de estrategia, selección de modelo de IA, entrenamiento y selección de modelo y optimización del modelo.



**Figura 4.2:** Diagrama del proceso para generar un modelo optimizado para pronosticar series de tiempo de la velocidad del viento

Durante la etapa de **selección de estrategia** se define cómo el modelo actuará con los datos para obtener pronósticos. Las estrategias utilizadas en este trabajo son: Directa, Recursiva, MIMO, DirREC y DIRMO.

Posteriormente, durante la etapa de **selección de modelo** se define el tipo de modelo de Inteligencia Artificial que se utilizará, siendo importante ya que define el formato que los datos deberán tener, ya sean vectores de datos o imágenes, en caso de utilizar Redes Neuronales Convolucionales (CNN).

Posteriormente se realiza el **preprocesamiento de datos**, donde se busca eliminar datos irregulares y adaptar a vectores o imágenes dado el modelo y estrategia que se implementa. Finalmente, se realiza el entrenamiento del modelo, y ajuste del mismo para buscar obtener resultados con mejores aptitudes, sean ajustes manuales o utilizando técnicas para optimización de hiperparámetros, siendo utilizados en el trabajo presente los algoritmos genéticos y búsqueda aleatoria.

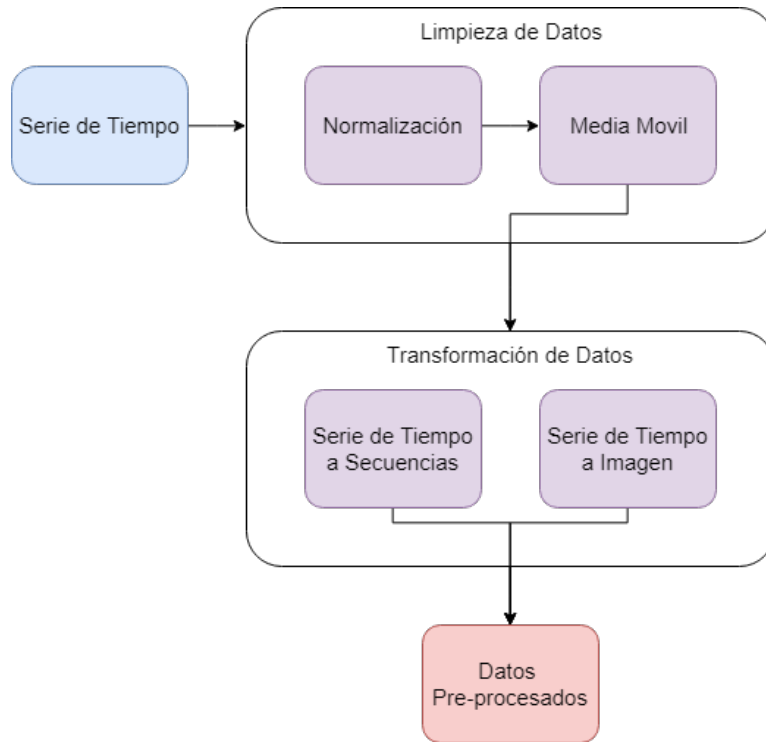
## 4.1. Preprocesamiento de datos

Durante el preprocesamiento se realizan múltiples tareas para preparar los datos previo al proceso de entrenamiento del modelo pronosticador para que estos puedan utilizarse de forma correcta. La Figura 4.3 muestra el proceso y subprocesos que se realizan durante el preprocesamiento de los datos, el cual consta dos procesos principales; **limpieza de datos** y **transformación de Datos**.

El proceso de Limpieza de Datos consiste en explorar la serie de tiempo para buscar datos atípicos, los cuales pueden impactar de forma negativa el proceso y causan resultados con un mayor error (Tsay, 1988). Posteriormente los datos deben transformarse a un formato que será utilizado por el modelo pronosticador correspondiente. En este trabajo se tienen dos principales formatos utilizados: transformación de serie de tiempo a secuencias y transformación de serie de tiempo a imágenes.

### 4.1.1. Limpieza de datos

La calidad de los datos puede afectar de manera negativa el resultado que se obtiene del modelo pronosticador que haya utilizado estos datos durante su proceso de entrenamiento. Estos datos se pueden originar debido a fallas en los equipos sensores, transmisores u otras irregularidades físicas presentes (Montgomery et al. 2015).



**Figura 4.3:** Diagrama de preprocesamiento de datos

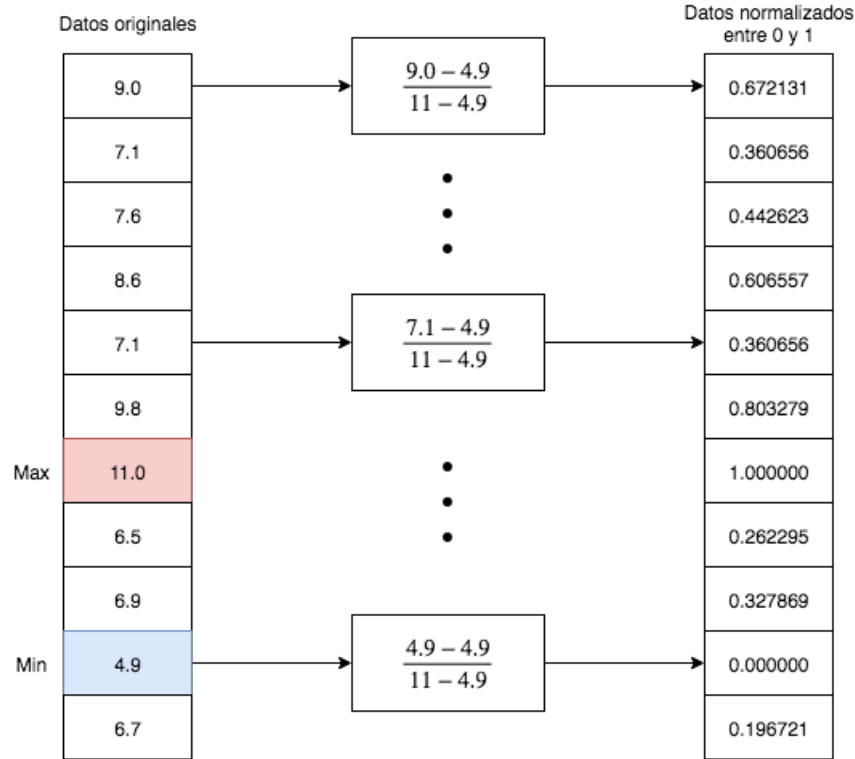
El proceso de **limpieza de datos** tiene como finalidad reducir el ruido en los datos e identificar y remover datos atípicos. Para este trabajo se utilizan las técnicas de normalización de datos y media móvil.

#### 4.1.1.1. Normalización de datos

El objetivo de la normalización de datos es garantizar que los rangos de los valores existentes dentro de la serie de tiempo no sea demasiado amplio. Esto ayuda a que el proceso de entrenamiento del modelo pronosticador sea más rápido, además de ayudar a reducir el posible sesgo que puede generar en el modelo de Inteligencia Artificial (Nayak et al. 2014). El rango utilizado en este trabajo es de  $[0, 1]$  y su expresión matemática se muestra en la Ecuación 4.1.

$$N_i = \frac{n_i - \min_n}{\max_n - \min_n} \quad (4.1)$$

donde  $N_i$  es el valor normalizado obtenido,  $n_i$  es el valor original de la serie a normalizar,  $min_n$  es el valor mínimo dentro de la serie y  $max_n$  es el valor máximo. La Figura 4.4 muestra el proceso realizado para la normalización de los datos para una serie de tiempo.



**Figura 4.4:** Proceso visual de normalización de datos para una serie de tiempo

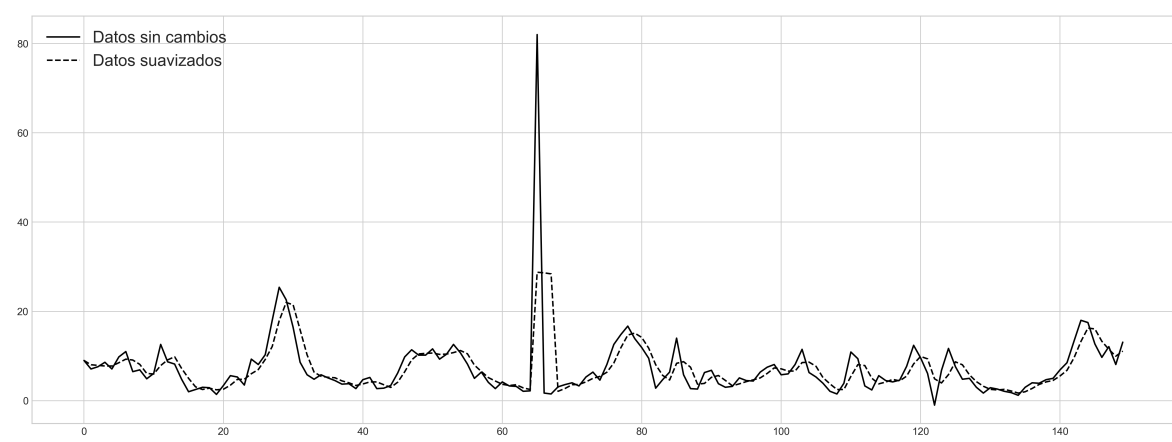
#### 4.1.1.2. Media Móvil

La media móvil es una función para suavizar o filtrar los datos, ya que reemplaza cada observación  $y_t$  con una combinación lineal de los otros datos que están cercanos en el tiempo (Montgomery et al. 2015). Una media móvil simple de rango  $N$  asigna pesos  $1/N$  para las  $N$  observaciones mas recientes  $Y_T, Y_{T-1}, \dots, Y_{T-N+1}$  y asigna cero a todas las demás observaciones, como define la Ecuación 4.2.

$$M_T = \frac{Y_T, Y_{T-1}, \dots, Y_{T-N+1}}{N} = \frac{1}{N} \sum_{t=T-N+1}^T y_t \quad (4.2)$$

donde  $M_T$  es el la observación nueva obtenida,  $Y$  las observaciones cercanas y  $N$  el

rango de las observaciones. Claramente las observaciones nuevas reemplazan las anteriores, añadiéndose a la siguiente operación (Montgomery et al., 2015). En la Figura 4.5 se puede observar un ejemplo de una fracción de serie de tiempo con suavizado por medio de media móvil.



**Figura 4.5:** Gráfica de muestra de media móvil en una serie de tiempo

## 4.1.2. Transformación de datos

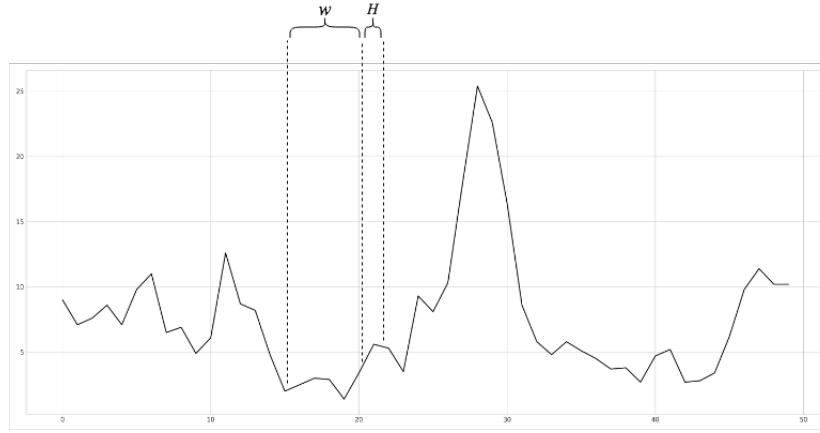
El proceso de transformación de datos es de gran importancia, ya que dará el formato adecuado para el proceso de entrenamiento del modelo pronosticador. Para este trabajo se puede dividir en dos tipos principales de datos, que son generar secuencias y generar imágenes.

### 4.1.2.1. Transformación de serie de tiempo a secuencias

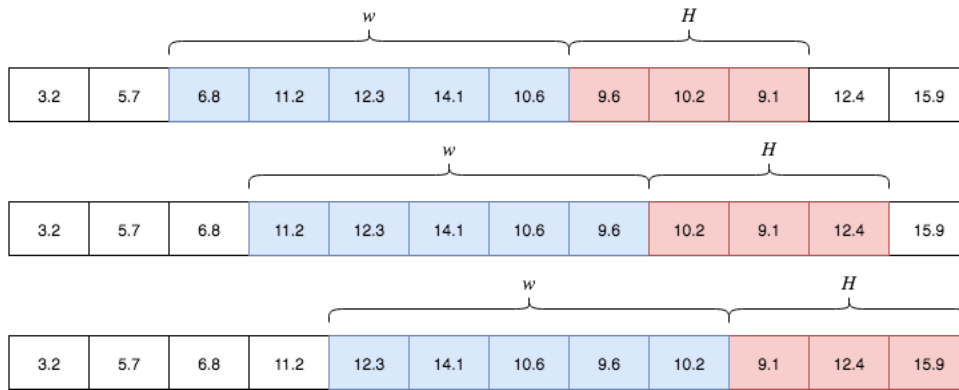
El proceso de transformación de una serie de tiempo a secuencias consiste en tomar la serie de tiempo (Figura 4.6), y generar múltiples secuencias que serán utilizadas por los modelos de IA durante el proceso de entrenamiento.

Como ilustra la Figura 4.7, cada secuencia de entrenamiento tiene longitud  $w + H$ , donde  $w$  son los datos de entrada y  $H$  datos esperados. Durante el proceso de entrenamiento, el  $w$  se considera un hiperparámetro que puede ser ajustado, y afecta el resultado obtenido por el modelo pronosticador, mientras que  $H$  es el horizonte máximo definido (24 horas para este trabajo).





**Figura 4.6:** Gráfica de muestreo de una serie de tiempo para extraer secuencia y dato de horizonte



**Figura 4.7:** Ejemplo de generación de secuencias a partir de una serie de tiempo

En la Tabla 4.1 se muestra el resultado esperado que se obtiene al convertir una serie de tiempo a secuencias. El total de secuencias generadas depende del tamaño de la serie de tiempo y el tamaño definido de ventana  $w$  y tamaño de horizonte  $H$ .

**Tabla 4.1:** Ejemplo de generación de secuencias

# Secuencia	$w$					<i>Horizonte Esperado</i>			
1	$\tilde{x}_1$	$\tilde{x}_2$	$\tilde{x}_3$	$\tilde{x}_4$	...	$\tilde{x}_w$	$\tilde{x}_{w+1}$	...	$\tilde{x}_{w+H}$
2	$\tilde{x}_2$	$\tilde{x}_3$	$\tilde{x}_4$	$\tilde{x}_5$	...	$\tilde{x}_{w+1}$	$\tilde{x}_{w+2}$	...	$\tilde{x}_{w+H+1}$
3	$\tilde{x}_3$	$\tilde{x}_4$	$\tilde{x}_5$	$\tilde{x}_6$	...	$\tilde{x}_{w+2}$	$\tilde{x}_{w+3}$	...	$\tilde{x}_{w+H+2}$
4	$\tilde{x}_4$	$\tilde{x}_5$	$\tilde{x}_6$	$\tilde{x}_7$	...	$\tilde{x}_{w+3}$	$\tilde{x}_{w+4}$	...	$\tilde{x}_{w+H+3}$
5	$\tilde{x}_5$	$\tilde{x}_6$	$\tilde{x}_7$	$\tilde{x}_8$	...	$\tilde{x}_{w+4}$	$\tilde{x}_{w+5}$	...	$\tilde{x}_{w+H+4}$
6	$\tilde{x}_6$	$\tilde{x}_7$	$\tilde{x}_8$	$\tilde{x}_9$	...	$\tilde{x}_{w+5}$	$\tilde{x}_{w+6}$	...	$\tilde{x}_{w+H+5}$
⋮			⋮					⋮	
$n$	$\tilde{x}_n$	$\tilde{x}_{n+1}$	$\tilde{x}_{n+2}$	$\tilde{x}_{n+3}$	...	$\tilde{x}_{w+n}$	$\tilde{x}_{w+n+1}$	...	$\tilde{x}_{w+n+H+1}$

#### 4.1.2.2. Transformación de de serie de tiempo a imagen

Para obtener una imagen a partir de la serie de tiempo, se utilizó la técnica de *Gramian Angular Field* (GAF) (Wang & Oates, 2015). Los datos de la serie de tiempo son transformados a coordenadas polares utilizando el coseno angular (*arccos*) a cada valor respectivo de la serie como muestra la Ecuación 4.3, y el paso de tiempo entre el número de datos para obtener su radio(*r*) por utilizando la Ecuación 4.4.

$$\phi_i = \arccos(\tilde{x}_i) \quad (4.3)$$

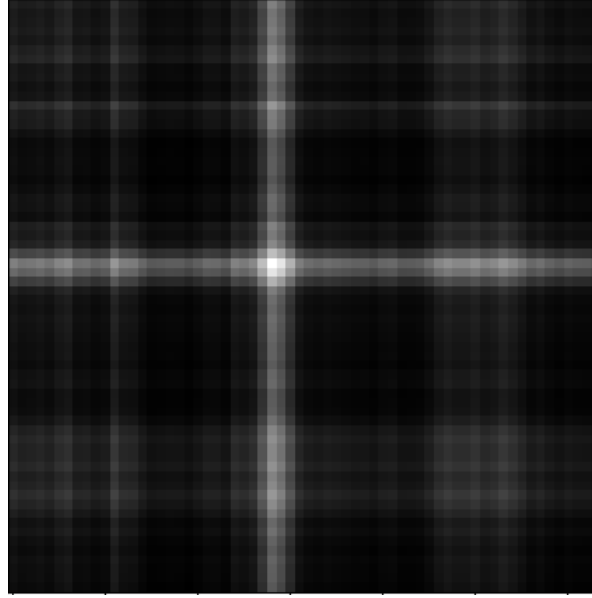
$$r_i = \frac{t_i}{N} \quad (4.4)$$

Para identificar la correlación temporal dentro de diferentes intervalos de tiempo se considera la suma trigonométrica entre cada punto (Wang & Oates, 2015). Esto se codifica en la matriz de Gram, la cual se define por la Ecuación 4.5.

$$G = \begin{bmatrix} \cos(\phi_1 + \phi_1) & \dots & \cos(\phi_1 + \phi_n) \\ \cos(\phi_2 + \phi_1) & \dots & \cos(\phi_2 + \phi_n) \\ \vdots & \ddots & \vdots \\ \cos(\phi_n + \phi_1) & \dots & \cos(\phi_n + \phi_n) \end{bmatrix}. \quad (4.5)$$

Entre otras propiedades, la matriz de Gram conserva la dependencia temporal de la serie de tiempo. Finalmente, la matriz se convierte en imagen en escala de gris con un escalamiento entre  $[0, 255]$ . La Figura 4.8 muestra un ejemplo de una serie de tiempo de 64 datos que se convierte a imagen.

Así como la transformación de series de tiempo a secuencias genera múltiples secuencias, la transformación de series de tiempo a imágenes genera múltiples imágenes. Cada imagen tiene asignado su horizonte (*h*) esperado. En la Tabla 4.2 se muestra un ejemplo de lógica que se obtiene al generar un banco de imágenes.



**Figura 4.8:** Ejemplo de Serie de Tiempo a Imagen tamaño de ventana 64

**Tabla 4.2:** Tabla de ejemplo de generación de Imágenes a partir de secuencias de serie de tiempo

# Imagen	$w$						<i>Horizonte Esperado</i>		
1	$\tilde{x}_1,$	$\tilde{x}_2,$	$\tilde{x}_3,$	$\tilde{x}_4,$	$\dots$	$\tilde{x}_w$	$\tilde{x}_{w+1},$	$\dots$	$\tilde{x}_{w+H}$
2	$\tilde{x}_2,$	$\tilde{x}_3,$	$\tilde{x}_4,$	$\tilde{x}_5,$	$\dots$	$\tilde{x}_{w+1}$	$\tilde{x}_{w+2},$	$\dots$	$\tilde{x}_{w+H+1}$
3	$\tilde{x}_3,$	$\tilde{x}_4,$	$\tilde{x}_5,$	$\tilde{x}_6,$	$\dots$	$\tilde{x}_{w+2}$	$\tilde{x}_{w+3},$	$\dots$	$\tilde{x}_{w+H+2}$
4	$\tilde{x}_4,$	$\tilde{x}_5,$	$\tilde{x}_6,$	$\tilde{x}_7,$	$\dots$	$\tilde{x}_{w+3}$	$\tilde{x}_{w+4},$	$\dots$	$\tilde{x}_{w+H+3}$
5	$\tilde{x}_5,$	$\tilde{x}_6,$	$\tilde{x}_7,$	$\tilde{x}_8,$	$\dots$	$\tilde{x}_{w+4}$	$\tilde{x}_{w+5},$	$\dots$	$\tilde{x}_{w+H+4}$
6	$\tilde{x}_6,$	$\tilde{x}_7,$	$\tilde{x}_8,$	$\tilde{x}_9,$	$\dots$	$\tilde{x}_{w+5}$	$\tilde{x}_{w+6},$	$\dots$	$\tilde{x}_{w+H+5}$
$\vdots$			$\vdots$					$\vdots$	
$n$	$\tilde{x}_n,$	$\tilde{x}_{n+1},$	$\tilde{x}_{n+2},$	$\tilde{x}_{n+3},$	$\dots$	$\tilde{x}_{w+n},$	$\tilde{x}_{w+n+1},$	$\dots$	$\tilde{x}_{w+n+H+1}$

## 4.2. Implementación de Estrategias de predicción múltiples pasos en el tiempo

Cuando se busca realizar pronósticos en series de tiempo en múltiples pasos en tiempo futuro se encuentra que la dificultad es considerablemente mayor que cuando se busca un pronóstico con un solo paso en el tiempo al futuro. Esto se debe a la acumulación de errores, reducción de precisión e incremento de incertidumbre (An & Anh, 2015). Por este motivo se han propuesto el uso de estrategias que buscan mejorar la aptitud de los pronósticos de series

de tiempo; Recursiva, Directa, MIMO, DIRMO y DirREC.

### 4.2.1. Estrategia Recursiva

El Algoritmo 4.1 se desarrollo para obtener predicciones con un modelo basado en estrategia Recursiva. Este algoritmo muestra la lógica utilizada en este trabajo para realizar pronósticos con la estrategia Recursiva por medio de múltiples iteraciones. El algoritmo recibe el modelo, los datos de entrada (*data*), el horizonte  $H$  y el tamaño de ventana de datos  $w$  (*window\_lenght*). El proceso consiste en una cola (*queue*) iniciando con  $w$ (tamaño de ventana) cantidad de datos(*data*) de la velocidad del viento que se utilizan para realizar el pronóstico. Posteriormente se inicializa la iteración, cuya condición es ciclar el horizonte a pronosticar, desde el punto inicial  $h = 0$  hasta el horizonte  $H$  máximo, avanzando un solo paso por ciclo. Dentro de la iteración se realiza la predicción con los datos extraídos de la cola y el modelo (*model*), los datos que se obtienen de regreso se almacenan en un arreglo *predictions* y se actualiza la cola con los datos que fueron pronosticados por el modelo. Una vez concluido el ciclo, se regresan los datos de pronóstico de la velocidad del viento.

---

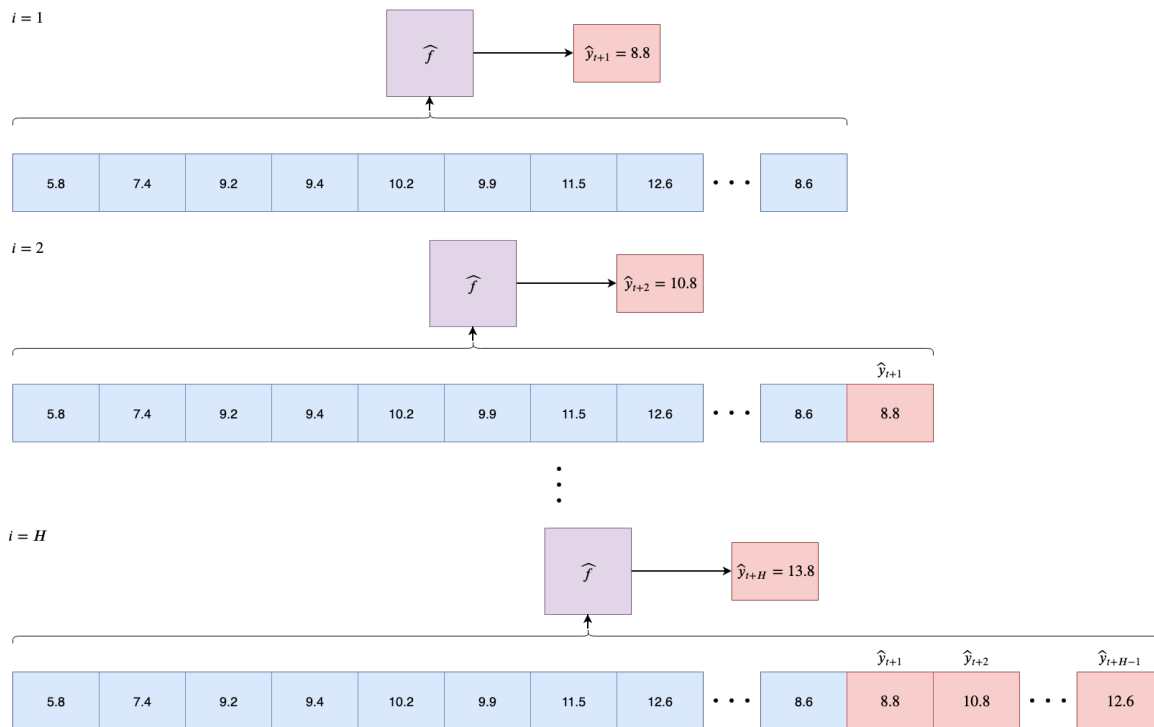
**Algoritmo 4.1** Algoritmo para Estrategia Recursiva

---

```
1: procedure RECURSIVE_STRATEGY(model, data, horizon, window_length)
2:   queue = data
3:   predictions = []
4:   for  $h = 0; 0 \leq horizon; h = h + 1$  do
5:     current = queue[-window_length :]
6:     predicted = model.predict(current)
7:     queue = queue.append(predicted)
8:     predictions.add(predicted)
9:   end for
   return predictions
10: end procedure
```

---

La Figura 4.9 ilustra el proceso que sigue un modelo pronosticador que utiliza la estrategia Recursiva.



**Figura 4.9:** Proceso visual que realiza un pronosticador con estrategia Recursiva

### 4.2.2. Estrategia Directa

En el Algoritmo 4.2 se puede observar la lógica de programación utilizada para realizar pronósticos con la estrategia Directa. La lógica utilizada es muy similar a la empleada por la estrategia Recursiva, donde la mayor diferencia es el pronóstico para cada horizonte con un modelo distinto. El Algoritmo recibe una colección que contiene los modelos (*models*), los datos de la velocidad del viento que serán utilizados para realizar el pronóstico (*data*) y el horizonte de pronóstico (*horizon*). El primer paso fue generar un arreglo (*predictions*) vacío el cual tiene el propósito de almacenar los datos pronosticados. En seguida, se inicia el proceso de la iteración, el cual tiene como tarea generar un ciclo para realizar el pronóstico para cada horizonte de interés. Dentro de esta iteración, el primer paso es extraer el modelo (*model*) de interés para el horizonte  $h$ , posteriormente este modelo es utilizado para obtener el pronóstico de este paso y almacenarlo en el arreglo de *predictions*. Al realizar el pronóstico de cada paso en el horizonte la iteración es finalizada y algoritmo regresa el arreglo *predictions* con los valores pronosticados.

---

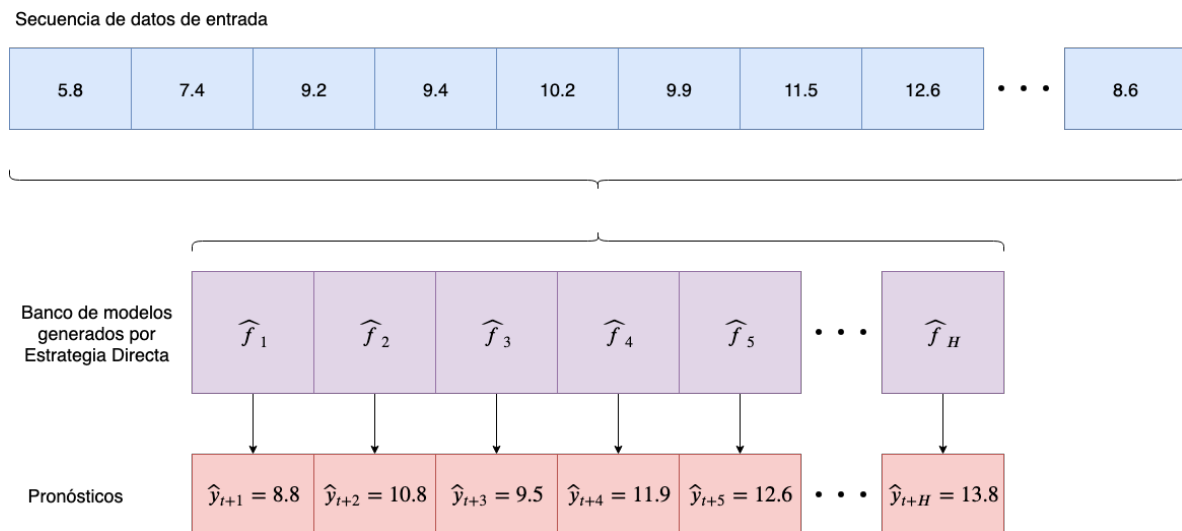
**Algoritmo 4.2** Algoritmo para Estrategia Directa

---

```
1: procedure DIRECT_STRATEGY(models, data, horizon)
2:   predictions = []
3:   for  $h = 0; 0 \leq horizon; h = h + 1$  do
4:     model = models[ $h$ ]
5:     predicted = model.predict(data)
6:     predictions.add(predicted)
7:   end for
8:   return predictions
9: end procedure
```

---

En la Figura 4.10 muestra el proceso realizado por un modelo pronosticador que utiliza la estrategia Directa.

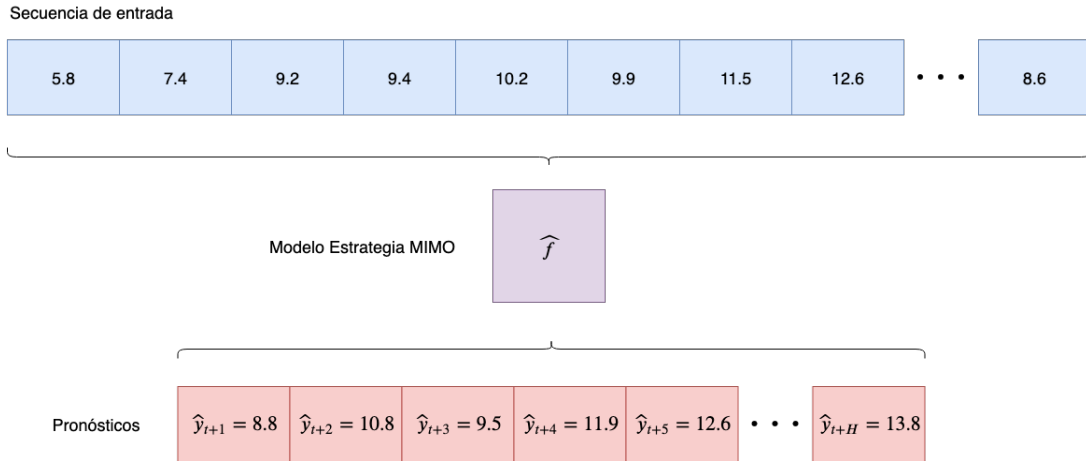


**Figura 4.10:** Proceso visual que realiza un pronosticador con estrategia Directa

### 4.2.3. Estrategia MIMO

La Estrategia MIMO se caracteriza por solamente utilizar un modelo capaz de pronosticar múltiples pasos en el tiempo. En la Figura 4.11 se puede observar que se toman  $w$  datos de entrada y se obtienen  $H$  salidas, que son los múltiples pasos en el tiempo que el modelo pronostica.

La Estrategia Directa es simple en implementación, pero puede generar alta dependencia entre las múltiples salidas que este entrega.



**Figura 4.11:** Proceso visual que realiza un pronosticador con estrategia MIMO

#### 4.2.4. Estrategia DIRMO

La Estrategia DIRMO combina la estrategia Directa y MIMO, donde se entrenan  $n$  modelos MIMO de  $s$  tamaño. En la Figura 4.12 se muestra un ejemplo para realizar pronósticos con un modelo construido con la Estrategia DIRMO donde  $n = 4$ , por lo que el pronóstico del horizonte ( $H$ ) se divide 4 bloques. La cantidad de pasos en el tiempo que pronostica cada uno de estos bloques se da por  $s = \frac{H}{n}$ , donde  $H$  y  $n$  son números enteros y el resultado de  $s$  debe ser un número entero, caso contrario uno o mas de un bloque deberá pronosticar una cantidad desigual de pasos en el tiempo.

En el Algoritmo 4.3 se observa la simple lógica utilizada para realizar pronóstico con la Estrategia DIRMO. El proceso se inicia cuando la función recibe una colección que contiene los  $n$  modelos que conforman el modelo completo y una colección que contiene las  $n$  cantidad

---

#### Algoritmo 4.3 Algoritmo para Estrategia DIRMO

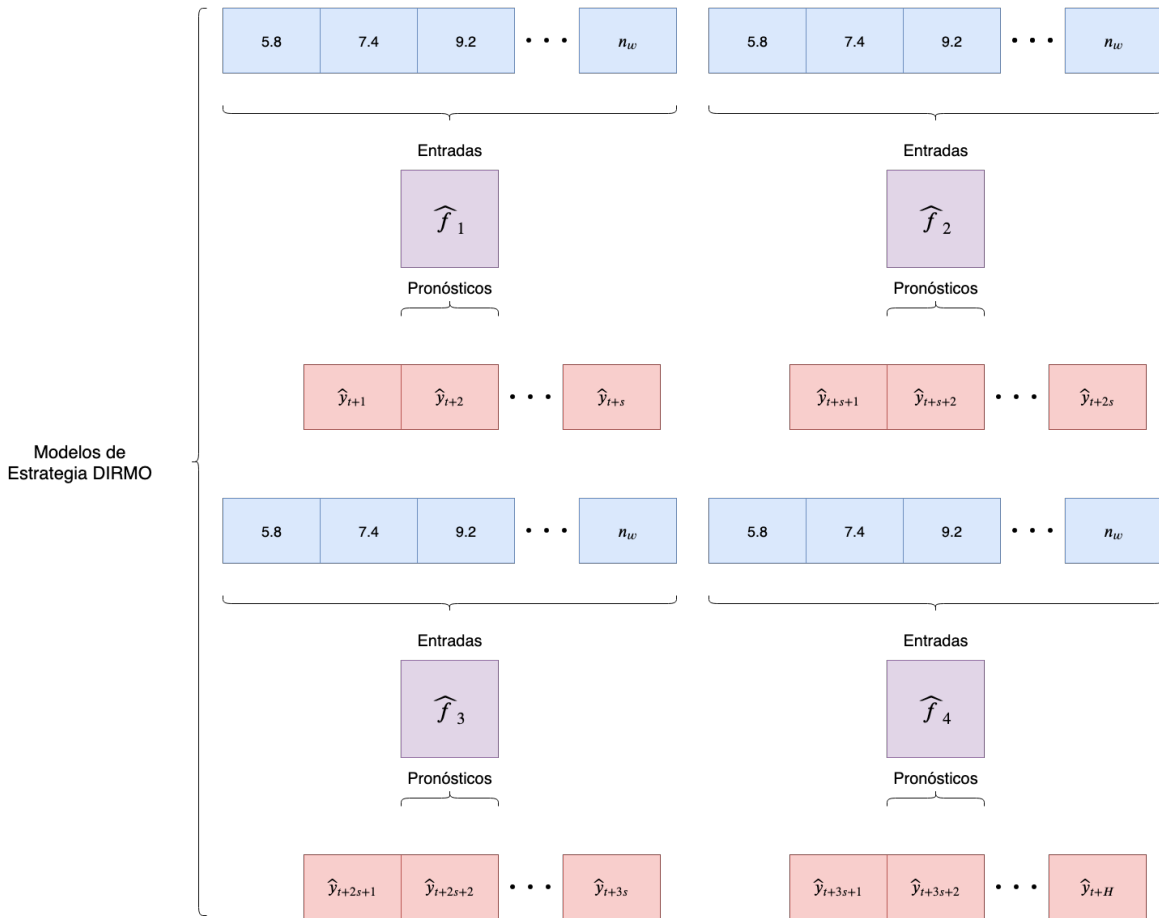
---

```

1: procedure DIRMO_STRATEGY(models, inputs, n)
2:   predictions = []
3:   for  $i = 0; 0 \leq n; i = i + 1$  do
4:     model = models[ $i$ ]
5:     input = inputs[ $i$ ]
6:     prediction = model.predict(data)
7:     predictions.add(prediction)
8:   end for
9:   return predictions
10: end procedure

```

---



**Figura 4.12:** Proceso visual que realiza un pronosticador con estrategia DIRMO

de entradas (*inputs*) que corresponden a cada uno de estos modelos. Finalmente, se realiza una iteración para obtener el pronóstico correspondiente con cada modelo y se regresa el pronóstico completo de  $H$  pasos en el tiempo.

#### 4.2.5. Estrategia DirREC

La estrategia DirREC es una combinación de las estrategias Directa y Recursiva. Esta estrategia busca eliminar la independencia generada por la estrategia Directa mediante el uso de la estrategia Recursiva. En este proceso se realiza retroalimentación del valor pronosticado de cada modelo a cada modelo subsecuente. En la Figura 4.13 se muestra el proceso que se realiza para  $H$  horizonte.

El Algoritmo 4.4 muestra la lógica utilizada para realizar un pronóstico de  $H$  horizonte. Este algoritmo es similar al utilizado para realizar pronósticos con la estrategia Recursiva, con



---

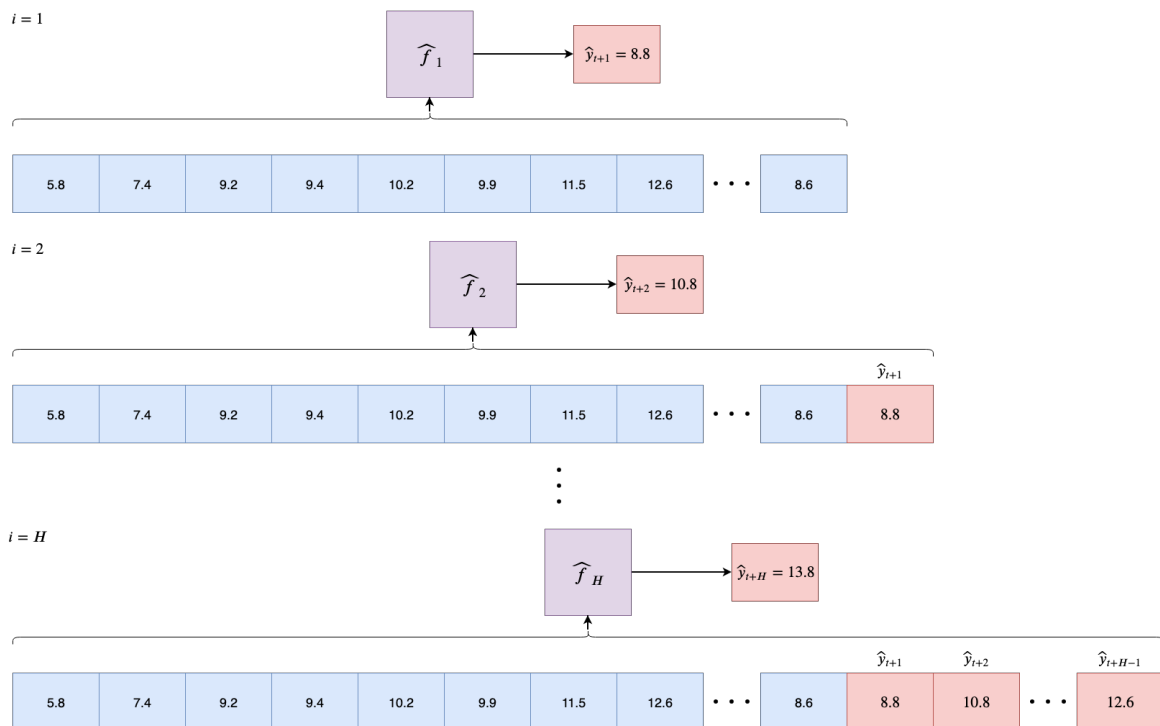
**Algoritmo 4.4** Algoritmo para Estrategia DirREC

---

```
1: procedure DIRREC_STRATEGY(models, data, horizon, window_length)
2:   queue = data
3:   predictions = []
4:   for  $h = 0; 0 \leq horizon; h = h + 1$  do
5:     model = models[h]
6:     current = queue[ $-window\_length :$ ]
7:     predicted = model.predict(current)
8:     queue = queue.append(predicted)
9:     predictions.add(predicted)
10:  end for
11:  return predictions
```

---

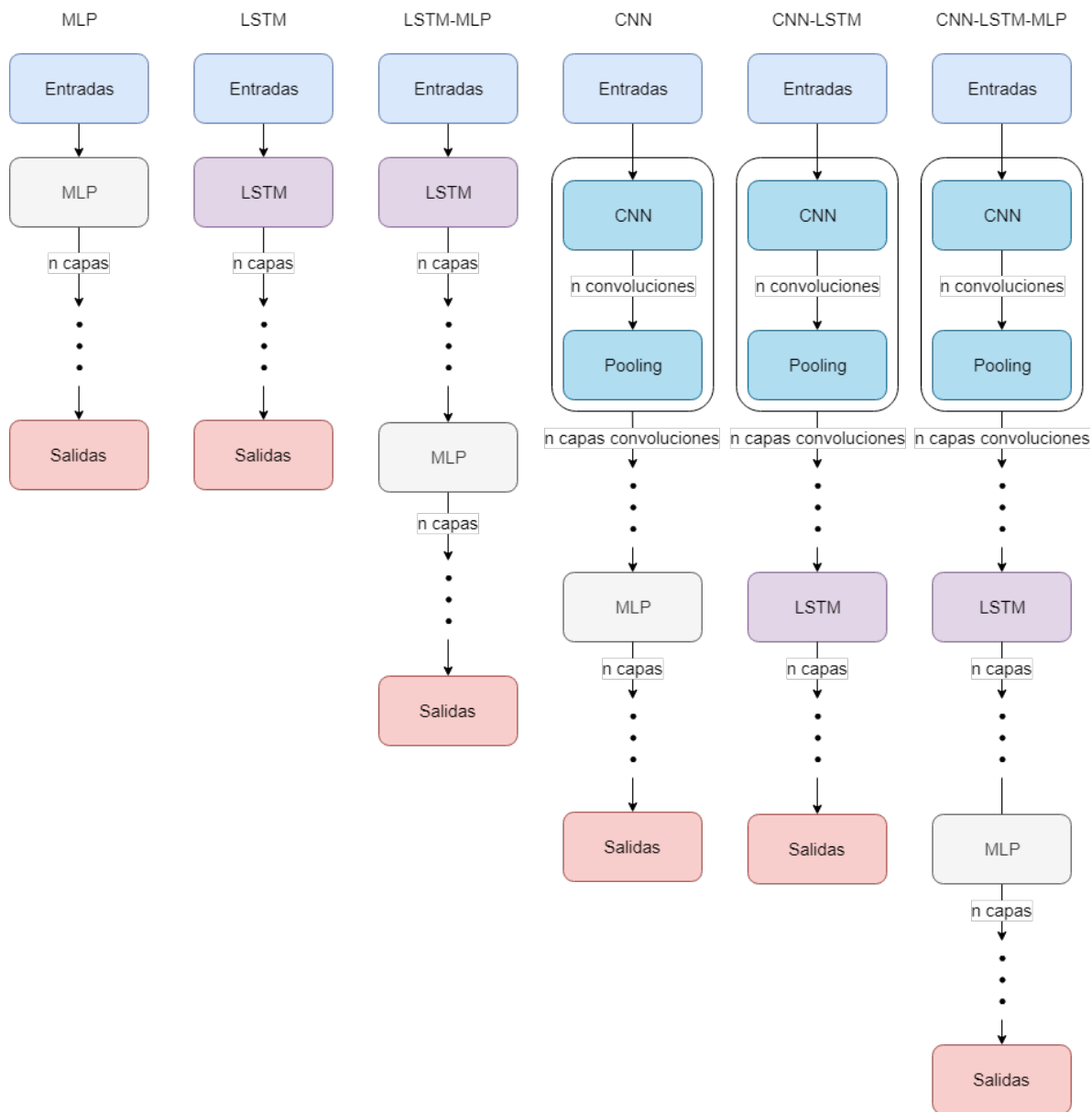
la diferencia de que cada pronóstico individual se realizará con un modelo independiente. Al igual que la estrategia Directa esta estrategia sufre por el poder computacional requerido o esfuerzo humano para realizar la optimización de los modelos, y similarmente a la estrategia Recursiva sufre de acumulación de errores a mayor horizonte de predicción utilizado.



**Figura 4.13:** Proceso visual que realiza un modelo con la estrategia DirREC

### 4.3. Modelos de Inteligencia Artificial

Para este trabajo se utilizaron 3 modelos de IA base; ANN, LSTM y CNN. Estos fueron combinados (Figura 4.14) para buscar obtener resultados con mayor aptitud al combinar las fortalezas que estos proveen. Los modelos de IA utilizados poseen dos tipos principales de hiperparámetros ajustables: globales y específicos.



**Figura 4.14:** Combinación de modelos de Inteligencia Artificial utilizados para realizar pronóstico de la velocidad del viento

Los **hiperparámetros globales** son aquellos que no son específicos a un solo modelo de IA, y que es común entre distintos modelos. Entre estos se encuentran:

- **Tamaño de ventana ( $w$ ):** Es la cantidad de datos históricos que se envían a las entradas del modelo pronosticador.
- **Épocas de entrenamiento o *Epochs*** por su nombre en inglés: Es el número de épocas, o ciclos durante los cuales el modelo de IA será entrenado. Este hiperparámetro aplica a los modelos de redes neuronales.
- **Tamaño de lote o por su nombre en inglés *batch size*:** Es el número de lote de datos de entrada que se procesan antes de que el modelo realice la actualización de sus parámetros internos.
- **Paciencia:** Este hiperparámetro es utilizado para detener el entrenamiento del modelo cuando no mejore en un número de épocas equivalente a la paciencia.
- **Rango de normalización:** Es el rango numérico utilizado para realizar la normalización de los datos, para este trabajo se utilizó un rango de  $[0, 1]$ ; sin embargo, es posible utilizar distintos rangos, tal como  $[-1, 1]$ . El rango más común es de  $[0, 1]$  debido a que reduce más el tiempo requerido por el modelo para alcanzar un punto de convergencia.

Los **hiperparámetros específicos** son aquellos que pertenecen a las propiedades que definen la construcción o parámetros de los modelos de IA. Entre estos se encuentran:

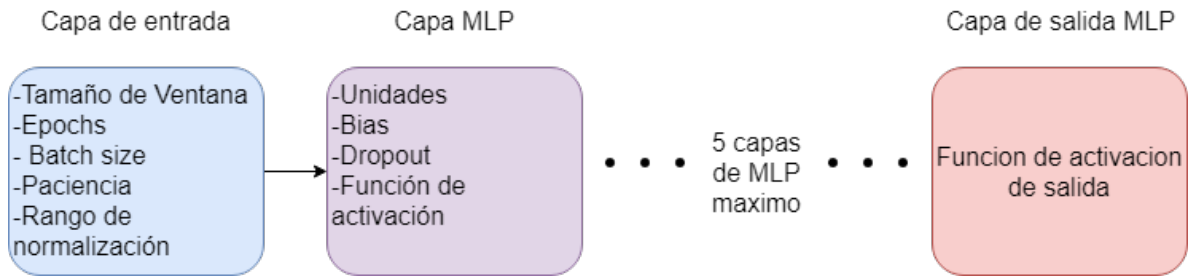
- **Número de Neuronas:** Es el número de neuronas que posee una ANN. Cada una de las capas que conforma una ANN puede tener un número distinto de neuronas.
- **Bias:** Este hiperparámetro es usualmente de tipo booleano (verdadero o falso), el cual tiene como función indicar si una capa o no tendrá una neurona conocida como sesgo, o por su nombre en inglés *bias*. Esta neurona no posee entrada previa y siempre entrega un "1" a las operaciones de la capa en la que se encuentra.
- **Bidireccional:** Hiperparámetro booleano que se utiliza para declarar si una capa de red neuronal recurrente utiliza retroalimentación en ambas direcciones.

- **Dropout:** El *dropout* es una técnica de regularización utilizada por ANN para reducir el sobre entrenamiento. El *dropout* reduce el valor de importancia que una neurona tiene, lo cual ayuda a dar prioridad a las neuronas con mayor impacto.
- **Dropout recurrente** (Solamente para LSTM): Es una técnica de regularización similar al *dropout* regular, sin embargo, este hiperparámetro es utilizado dentro de las redes neuronales recurrentes para controlar el valor de importancia entre los valores entregados por la retroalimentación de cada neurona recurrente.
- **Función de activación:** Es aquella que es utilizada en la salida de cada neurona, entre las cuales se encuentran las funciones: relu, elu, linear, y tanh.
- **Función de activación de salida:** Es similar a la función de activación, pero se utiliza específicamente en la capa de salida. Para este trabajo se utilizaron linear y sigmoide.
- **Relleno o por su nombre en inglés *padding*:** Este hiperparámetro define que tipo relleno es posible que el modelo CNN. Tiene como objetivo ayudar al kernel que toma valores de la matriz de datos en caso que este se salga de rango, debido a un tamaño irregular de matriz ó kernel.
- **Paso de kernel o por su nombre en inglés *stride*:** Es el paso de dato en el que se puede mover el kernel de un modelo CNN.

Los 6 modelos utilizados en este trabajo fueron limitados en su cantidad de neuronas máximas, número de capas, tamaño de ventana y variación entre los distintos hiperparámetros ajustables; esto con el objetivo de reducir el espacio de búsqueda requerido para ajustar cada uno de estos.

#### 4.3.1. Hiperparámetros de implementación del modelo MLP

El modelo implementado con MLP fue de los mas simples y con menor cantidad de hiperparámetros que requerían de ajustes. En la Figura 4.15 se muestra una representación en la cual se muestran los hiperparámetros utilizados por el modelo implementado con MLP.



**Figura 4.15:** Representación visual de la configuración para el modelo implementado con perceptrón multicapas (MLP)

Para este trabajo se limitó como un máximo el uso de 5 capas MLP, obteniéndose un total de 26 hiperparámetros máximos ajustables.

En la Tabla 4.3 se muestran las variaciones posibles que se utilizaron por cada hiperparámetro para la implementación del modelo MLP.

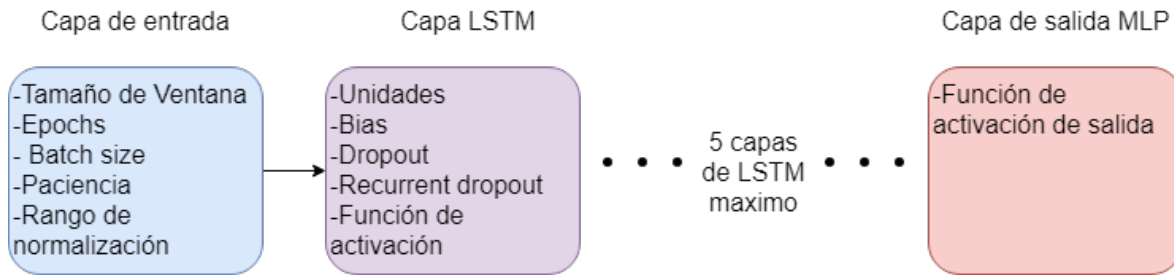
**Tabla 4.3:** Hiperparámetros utilizados para la configuración del modelo MLP implementado

<b>Hiperparámetro globales</b>	<b>Variaciones</b>
Tamaño de ventana	4-128
Epochs	100-10000
Paciencia	4-16
Rango de normalización	[0,1] ó [-1,1]
<b>Hiperparámetro de capas MLP</b>	<b>Variaciones</b>
Unidades (No. Neuronas)	2-512
Bias	True — False
Dropout	0-0.9
Función de activación	relu, elu, linear ó tanh
<b>Hiperparámetro de capa de salida</b>	<b>Variaciones</b>
Función de activación de salida	linear ó sigmoid

### 4.3.2. Hiperparámetros de implementación del modelo LSTM

La implementación del modelo LSTM es muy similar a la implementación del modelo MLP, con la diferencia de que el modelo LSTM posee un hiperparámetro mas (*recurrent dropout*) en sus capas. En la Figura 4.16 se muestra la configuración posible con las redes LSTM. Para este trabajo se limitó igualmente el número de capas de LSTM a 5, obteniéndose un máximo de 31 hiperparámetros ajustables.

En la Tabla 4.4 se muestran el rango de variaciones dentro los cuales se configuro el modelo LSTM.



**Figura 4.16:** Representación visual de la configuración para el modelo implementado con redes de gran memoria a corto plazo (LSTM)

**Tabla 4.4:** Hiperparámetros utilizados para la configuración del modelo LSTM implementado

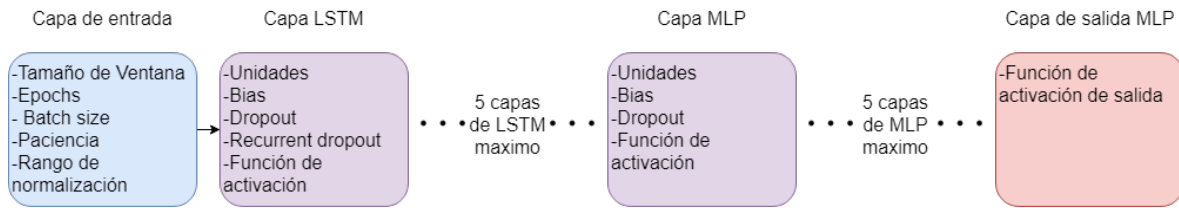
<b>Hiperparámetro globales</b>	<b>Variaciones</b>
Tamaño de ventana	4-128
Epochs	100-10000
Paciencia	4-16
Rango de normalización	[0,1] ó [-1,1]
<b>Hiperparámetro de capas LSTM</b>	<b>Variaciones</b>
Unidades (No. Neuronas)	2-512
Bias	True — False
Dropout	0-0.9
Recurrent dropout	0-0.5
Función de activación	relu, elu, linear o tanh
<b>Hiperparámetro de capa de salida</b>	<b>Variaciones</b>
Función de activación de salida	linear ó sigmoid

### 4.3.3. Hiperparámetros de implementación del modelo LSTM-MLP

La combinación de LSTM y MLP busca combinar el potencial de ambos modelos de IA, causando la necesidad de configurar ambos modelos de IA en un solo modelo pronosticador. El modelo se construye con capas de LSTM (1 a 5 capas) y MLP (1 a 5 capas), como se muestra en la Figura 4.17.

Los hiperparámetros que utiliza el modelo LSTM-MLP son una combinación de ambos, y similar a sus implementaciones individuales, se optó por utilizar 5 capas para cada uno como limitación. Para esta implementación se tiene un máximo de 51 hiperparámetros que se pueden combinar durante los ajustes de este modelo.

En la Tabla 4.5 se muestran las variaciones de hiperparámetros utilizados para la combinación de modelo LSTM y MLP.



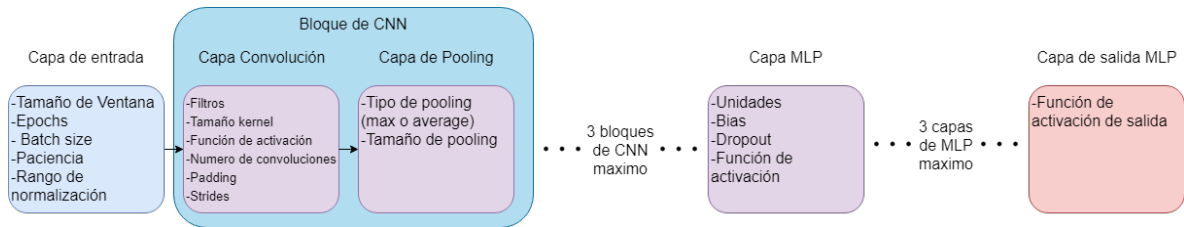
**Figura 4.17:** Representación visual de la configuración para el modelo implementado con redes de gran memoria a corto plazo y perceptrón multicapas (LSTM-MLP)

**Tabla 4.5:** Hiperparámetros utilizados para la configuración del modelo LSTM-MLP implementado

<b>Hiperparámetro globales</b>	<b>Variaciones</b>
Tamaño de ventana	4-128
Epochs	100-10000
Paciencia	4-16
Rango de normalización	[0,1] ó [-1,1]
<b>Hiperparámetro de capas LSTM</b>	<b>Variaciones</b>
Unidades (No. Neuronas)	2-512
Bias	True — False
Dropout	0-0.9
Función de activación	relu, elu, linear o tanh
Recurrent Dropout	0-0.5
<b>Hiperparámetro de capas MLP</b>	<b>Variaciones</b>
Unidades (No. Neuronas)	2-512
Bias	True — False
Dropout	0-0.9
Función de activación	relu, elu, linear o tanh
<b>Hiperparámetro de capa de salida</b>	<b>Variaciones</b>
Función de activación de salida	linear ó sigmoid

#### 4.3.4. Hiperparámetros de implementación del modelo CNN

El ajuste de hiperparámetros de la implementación del modelo CNN y sus variaciones son los que presentan mayor dificultad, esto debido a la gran cantidad de hiperparámetros y sus variaciones que son posibles de ajustar. Para este modelo se utiliza el término bloque para referirse a un bloque de convolución completo, el cual incluye las capas de convolución (pueden ser 2 por bloque) y la capa de pooling que siempre esta presente después de una convolución. El modelo se construye por bloque(s) de convolución, capas MLP y una capa de salida, como se muestra en la Figura 4.18.



**Figura 4.18:** Representación visual de la configuración para el modelo implementado redes neuronales convolucionales (CNN)

En la Tabla 4.6 se muestran los hiperparámetros utilizados para la configuración de modelos CNN implementados para este trabajo.

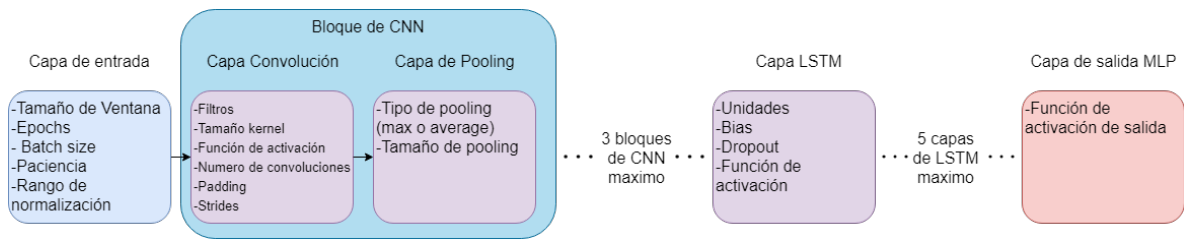
**Tabla 4.6:** Hiperparámetros utilizados para la configuración del modelo CNN implementado

<b>Hiperparámetro globales</b>	<b>Variaciones</b>
Tamaño de ventana	4-128
Epochs	100-10000
Paciencia	4-16
Rango de normalización	[0,1] ó [-1,1]
<b>Hiperparámetro de capas CNN</b>	<b>Variaciones</b>
Filtros	8-64
Tamaño de kernel	(3,3), (4,4), (5,5) y (6,6)
Padding	Same ó valid
Función de activación	relu ó elu
Stride	1-3
Número de convoluciones	1-2
<b>Hiperparámetros de capa de pooling</b>	<b>Variaciones</b>
Tipo de pooling	Average y Max
Tamaño de pooling	(2,2), (3,3) y (4,4)
<b>Hiperparámetro de capas MLP</b>	<b>Variaciones</b>
Unidades (No. Neuronas)	2-512
Bias	True — False
Dropout	0-0.9
Función de activación	relu, elu, linear o tanh
<b>Hiperparámetro de capa de salida</b>	<b>Variaciones</b>
Función de activación de salida	linear ó sigmoid

#### 4.3.5. Hiperparámetros de implementación del modelo CNN-LSTM

La combinación de los modelos CNN y LSTM presentan un gran potencial. Sin embargo, la complejidad y cantidad generada por la cantidad de hiperparámetros crece en comparación de modelos previos. Similar al modelo CNN, este modelo implementa de 1 a 5 capas de LSTM, generando un máximo de 55 hiperparámetros que son posibles de ajustar. En la Figura





**Figura 4.19:** Representación visual de la configuración para el modelo implementado redes neuronales convolucionales y redes de gran memoria a corto plazo (CNN-LSTM)

4.19 se muestra la forma en que se combinan los modelos de CNN y LSTM.

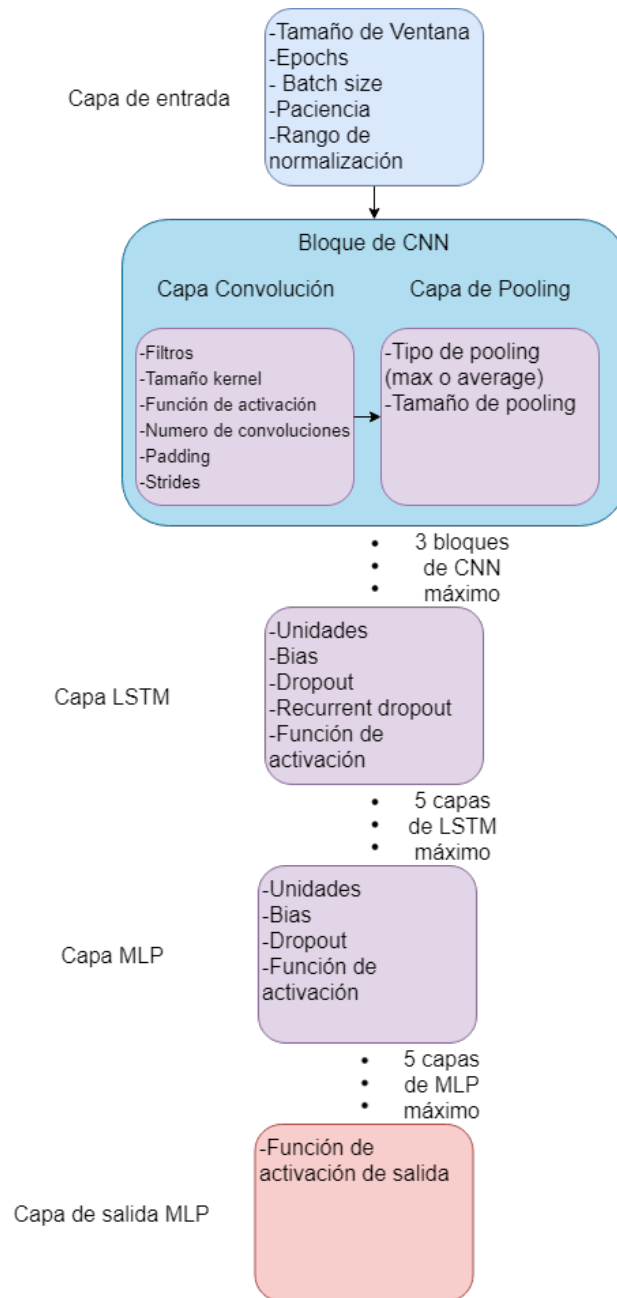
En la Tabla 4.7 se muestran los hiperparámetros específicos utilizados en el modelo CNN-LSTM.

**Tabla 4.7:** Hiperparámetros utilizados para la configuración del modelo CNN-LSTM implementado

<b>Hiperparámetro globales</b>	<b>Variaciones</b>
Tamaño de ventana	4-128
Epochs	100-10000
Paciencia	4-16
Rango de normalización	[0,1] ó [-1,1]
<b>Hiperparámetro de capas CNN</b>	<b>Variaciones</b>
Filtros	8-64
Tamaño de kernel	(3,3), (4,4), (5,5) y (6,6)
Padding	Same ó valid
Función de activación	relu ó elu
Stride	1-3
Número de convoluciones	1-2
<b>Hiperparámetros de capa de pooling</b>	<b>Variaciones</b>
Tipo de pooling	Average y Max
Tamaño de pooling	(2,2), (3,3) y (4,4)
<b>Hiperparámetro de capas LSTM</b>	<b>Variaciones</b>
Unidades (No. Neuronas)	2-512
Bias	True — False
Dropout	0-0.9
Recurrent dropout	0-0.5
Función de activación	relu, elu, linear o tanh
<b>Hiperparámetro de capa de salida</b>	<b>Variaciones</b>
Función de activación de salida	linear ó sigmoid

### 4.3.6. Hiperparámetros de implementación del modelo CNN-LSTM-ANN

Este modelo es el mas complejo y con mayor cantidad de hiperparámetros utilizados, esto debido a que combina los modelos de CNN, LSTM y ANN (Figura 4.20). En total este modelo utiliza 71 hiperparámetros como máximo.



**Figura 4.20:** Representación visual de la configuración para el modelo implementado redes neuronales convolucionales, redes de gran memoria a corto plazo y perceptrón multicapa (CNN-LSTM-MLP)

En la Tabla 4.8 se muestran los hiperparámetros y sus variaciones por capas que se ajustan para el modelo CNN-LSTM-MLP.

**Tabla 4.8:** Hiperparámetros utilizados para la configuración del modelo CNN-LSTM-MLP implementado

<b>Hiperparámetro globales</b>	<b>Variaciones</b>
Tamaño de ventana	4-128
Epochs	100-10000
Paciencia	4-16
Rango de normalización	[0,1] ó [-1,1]
<b>Hiperparámetro de capas CNN</b>	<b>Variaciones</b>
Filtros	8-64
Tamaño de kernel	(3,3), (4,4), (5,5) y (6,6)
Padding	Same ó valid
Función de activación	relu ó elu
Stride	1-3
Número de convoluciones	1-2
<b>Hiperparámetros de capa de pooling</b>	<b>Variaciones</b>
Tipo de pooling	Average y Max
Tamaño de pooling	(2,2), (3,3) y (4,4)
<b>Hiperparámetro de capas LSTM</b>	<b>Variaciones</b>
Unidades (No. Neuronas)	2-512
Bias	True — False
Dropout	0-0.9
Recurrent dropout	0-0.5
Función de activación	relu, elu, linear o tanh
<b>Hiperparámetro de capas MLP</b>	<b>Variaciones</b>
Unidades (No. Neuronas)	2-512
Bias	True — False
Dropout	0-0.9
Función de activación	relu, elu, linear o tanh
<b>Hiperparámetro de capa de salida</b>	<b>Variaciones</b>
Función de activación de salida	linear ó sigmoid

## 4.4. Optimización de hiperparámetros

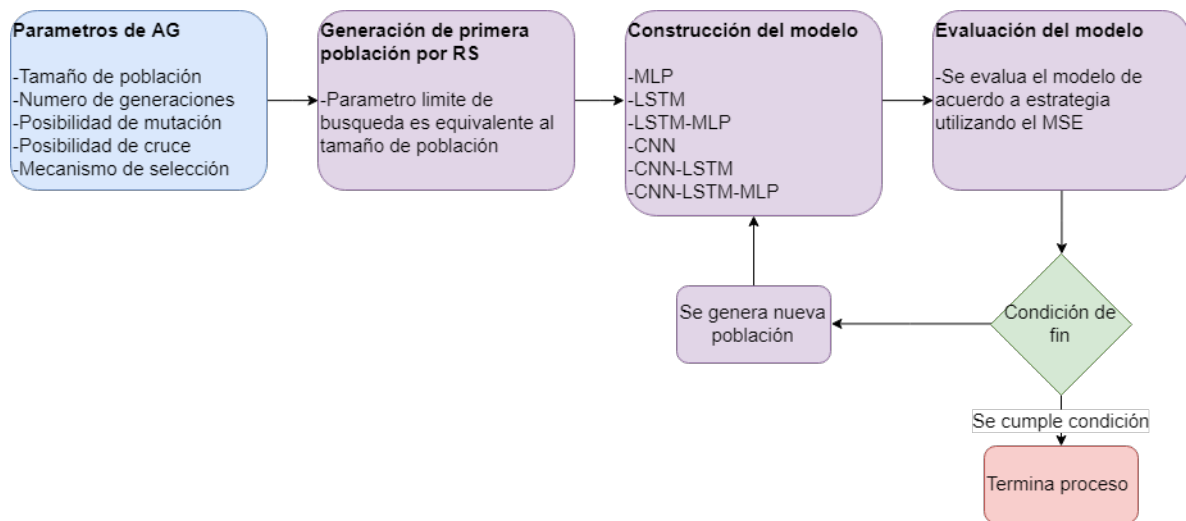
La tarea de ajustar hiperparámetros consume mucho tiempo al realizarse a prueba y error. Es por esto que en este trabajo se propone el uso de algoritmos genéticos (AG) y *random search* (RS) para la automatización de esta tarea.

Para este trabajo se utilizó el *framework* de computación evolutiva DEAP (Fortin et al., 2012) para la aplicación de AG simple en combinación con RS (Pedregosa et al., 2011) como iniciador de la primera generación de población de individuos.

El proceso consta de 4 etapas principales que son repetidos hasta llegar que la condición

programadas se cumpla. Las etapas utilizadas son:

1. Inicialización de parámetros para AG
2. Generación de primera generación mediante RS
3. Construcción del modelo
4. Evaluación del modelo



**Figura 4.21:** Esquema de algoritmo genético implementado

En la Figura 4.21 se muestra un esquema general del proceso seguido implementado para optimizar hiperparámetros. Los parámetros utilizados para este trabajo en el AG son los que se muestran en la Tabla 4.9.

**Tabla 4.9:** Parámetros utilizados para algoritmo genético

Parámetro	Valor
Tamaño de población	100
Número de generaciones	200
Posibilidad de mutación	30 %
Posibilidad de cruce	20 %
Mecanismo de selección	Selección por torneo 6 mejores

## 4.5. Tecnologías utilizadas

Este trabajo se realizó utilizando múltiples herramientas tecnológicas empleadas para el desarrollo de software, ambientes virtuales y librerías. Estas herramientas agilizaron el desarrollo de este trabajo lo cual hizo posible realizar múltiples experimentos para acertar la aptitud de los distintos modelos y estrategias de pronóstico. Entre las herramientas utilizadas durante el desarrollo de este trabajo destacan: Python, Tensorflow, Keras y Docker.

### 4.5.1. Python

El lenguaje de python es de alto nivel, interpretado para programación de uso general el cual soporta múltiples paradigmas (objetos, imperativa y funcional) (Oliphant, 2007). Este ha tenido un crecimiento en popularidad gracias a las librerías que facilitan el trabajo de análisis de datos e IA. Para este trabajo se utilizó la versión 3.6.6 de python.

### 4.5.2. Tensorflow

Tensorflow es un sistema de aprendizaje maquina que puede operar a gran escala y en un ambiente heterogéneo (Abadi et al., Abadi et al.). Esta librería trabaja con flujo de datos a través de grafos que representan estados y las operaciones que mutan estos estados. Tiene la facilidad de poder trabajar con CPU (*Central Processing Unit*), GPU (*Graphic Processing Unit*) y TPU (*Tensor Processing Unit*), además de incluir todos los modelos de aprendizaje profundo utilizados durante la elaboración de este trabajo.

### 4.5.3. Keras

Esta librería es una API (*Application Programming Interface*) que facilita el uso de aplicaciones de ML de mas bajo nivel (Gulli & Pal, 2017). Para este trabajo se utilizó Keras para facilitar la programación de los modelos que utilizan Tensorflow, agilizando el proceso al mismo tiempo que no se pierde flexibilidad, gracias a la modularidad que ofrece la API de Keras.

#### 4.5.4. Docker

Cuando se realizan trabajos de investigación que requieren el uso y configuración de múltiples equipos de cómputo, es fácil perder estas configuraciones y con ello perder la capacidad de poder reproducir los resultados obtenidos. Para este trabajo se utilizó Docker para crear una *imagen* que contiene todas las configuraciones utilizadas para realizar los experimentos. Docker es un conjunto de programas y servicios que se utilizan en grande escala para sistemas en la nube, sistemas distribuidos, entre otros; sin embargo, también es posible utilizar esta herramienta para facilitar los trabajos de investigación (Boettiger, 2015).

---

## Capítulo 5

# Análisis de Resultados

---

Para mostrar la efectividad de la metodología propuesta se realizaron múltiples experimentos con 5 diferentes series de tiempo de la velocidad del viento. Se utilizó una combinación entre los modelos de IA y estrategias de pronóstico a múltiples pasos. Estas series de tiempo fueron capturadas por estaciones climatológicas en distintas ubicaciones en el estado de Michoacán, México. El instrumento utilizado por las estaciones climatológicas para la captura de la velocidad del viento se llama anemómetro, el cual es de uso muy común en ubicaciones de interés para la captura de la velocidad del viento (Boyes, 2009). Estas series de tiempo se encuentran en intervalos horarios, es decir, las capturas que fueron tomadas dentro del rango de una hora son promediadas.

Las series de tiempo utilizadas presentan una gran cantidad de datos faltantes y datos atípicos, por que lo que a partir de estas series de tiempo originales se extrajeron subseries. Estas subseries de tiempo fueron extraídas al analizar y validar de forma manual secciones de la serie de tiempo original, las cuales son secciones que presentaron menor cantidad de datos faltantes ó datos atípicos.

En la Tabla 5.1 se muestran las distintas series de tiempo utilizadas además de su cantidad de datos totales. La distribución de los datos para entrenamiento y validación fue del 70 % y 30 % respectivamente.

Los experimentos realizados se muestran en las siguientes sub-secciones que corresponden a cada una de las series de tiempo utilizadas para este trabajo, donde se muestran los resultados obtenidos por los 7 distintos modelos y 5 estrategias de pronóstico. Además se muestran resultados obtenidos con la técnica de optimización de hiperparámetros para el mo-

**Tabla 5.1:** Cantidad de datos utilizados por cada una de las series de tiempo para entrenamiento y validación

Ubicación	Total de datos intervalo horario	Datos utilizados para entrenamiento	Datos utilizados para validación
Aristeo Mercado	21736	15215	6521
Cointzio	42364	29655	12709
Melchor Campo	27592	19314	8278
Corrales	20379	14265	6114
La Piedad	28973	20281	8692

delo con mejor aptitud.

Las métricas utilizadas para medir la aptitud de cada modelo y estrategia fueron: MSE, MAE, RMSE y tiempo de entrenamiento. El MSE es la métrica de mayor importancia debido a que fue empleada para ajustar los modelos, por lo que afecta directamente la aptitud del mismo. No obstante, el tiempo de entrenamiento también es una métrica de gran importancia, ya que se puede obtener un modelo con gran aptitud en MSE, pero un pésimo tiempo de entrenamiento. Es por esto que se debe considerar un balance entre precisión de modelo y costo computacional (el cual es directamente afectado por tiempo de entrenamiento) durante la selección del mejor modelo.

## 5.1. Resultados de Aristeo Mercado

Los resultados obtenidos durante los experimentos realizados con la serie de tiempo de Aristeo Mercado (ver Tabla 5.2) mostraron favorables resultados con la estrategia MIMO, particularmente con los modelos MLP y LSTM. Sin embargo, otras estrategias obtuvieron resultados similares y poseen una mayor cantidad de parámetros ajustables al poseer más de un solo modelo (Directa, DIRMO y DirREC). No obstante el tiempo que toma realizar un entrenamiento para los modelos es de gran importancia, esto debido a que el costo del modelo se incrementa al buscar la optimización de este modelo, donde se requiere entrenar miles de veces y unos cuantos segundos por modelo puede hacer la diferencia de días de entrenamiento. El mejor modelo es seleccionado utilizando la métrica del MSE, donde un menor número es considerado mejor. La Tabla 5.2 resalta los mejores resultados para cada una de las métricas utilizadas.

Durante este trabajo se realizó optimización de hiperparámetros para los modelos y estrategias que obtuvieron un mejor desempeño durante la fase de ajuste a prueba y error, esto



**Tabla 5.2:** Resultados obtenidos con los distintos modelos y estrategias utilizando la serie de tiempo de Aristeo Mercado

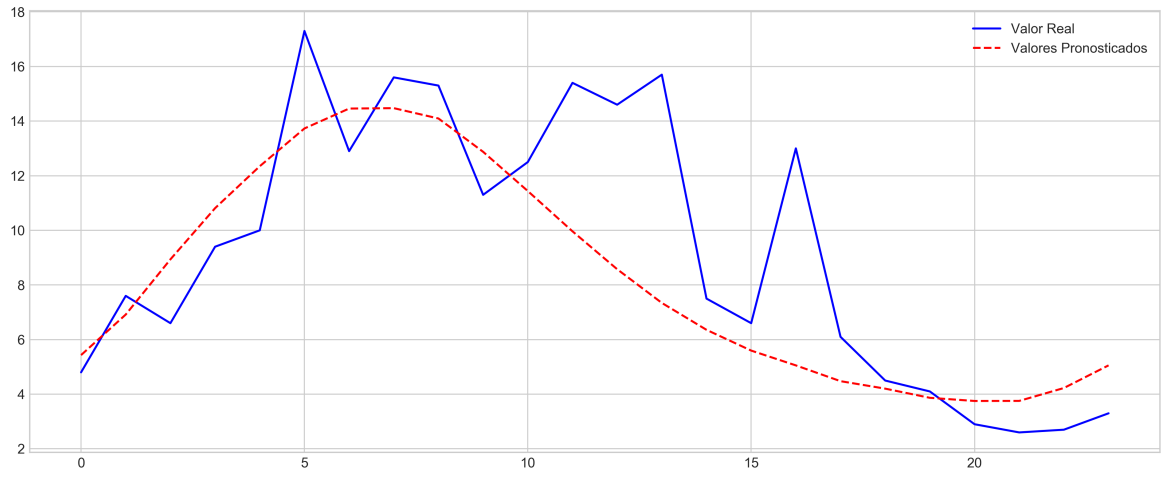
Modelo		k-NN	MLP	LSTM	LSTM-MLP	CNN	CNN-LSTM	CNN-LSTM-MLP
Estrategia de pronóstico	Métrica							
Estrategia Recursiva	MSE	0.00614	0.00495	<b>0.00435</b>	0.00466	0.00595	0.00631	0.00579
	MAE	0.06033	0.05316	<b>0.05020</b>	0.05116	0.05915	0.05670	0.05610
	RMSE	0.07838	0.07040	<b>0.06596</b>	0.06829	0.07718	0.07948	0.07612
	Tiempo	<b>0.0149s</b>	30.8s	65.4s	48.05s	6.48s	78.1s	60.2s
Estrategia Directa	MSE	0.00495	0.00412	<b>0.00382</b>	0.00398	0.00470	0.00434	0.00524
	MAE	0.05260	0.04845	<b>0.04543</b>	0.04652	0.05031	0.04781	0.0543
	RMSE	0.07033	0.06419	<b>0.06180</b>	0.06311	0.06859	0.06588	0.07243
	Tiempo	<b>12.08s</b>	181.14s	922.53s	1326.5s	1187.3s	1438s	1328s
Estrategia MIMO	MSE	0.00495	0.00371	<b>0.00370</b>	0.00380	0.00470	0.00444	0.00398
	MAE	0.05260	0.04468	<b>0.04329</b>	0.04596	0.05031	0.04851	0.04393
	RMSE	0.07033	0.06096	<b>0.06084</b>	0.06171	0.0686	0.06666	0.06315
	Tiempo	<b>0.015s</b>	43.9s	84.4s	102.1s	87.9s	134.3s	219.8s
Estrategia DIRMO	MSE	0.0082	0.0038	0.00412	<b>0.00379</b>	0.00616	0.00611	0.00646
	MAE	0.0877	0.04556	0.04852	<b>0.04514</b>	0.05984	0.05964	0.06582
	RMSE	0.0905	0.0616	0.06425	<b>0.0615</b>	0.07851	0.07818	0.08037
	Tiempo	<b>2.02s</b>	38.9s	636.61s	83.73s	83.02s	171.92s	182.83s
Estrategia DirREC	MSE	0.00495	0.00518	<b>0.00421</b>	0.00461	0.0059	0.0059	0.0058
	MAE	0.05260	0.05260	<b>0.04912</b>	0.05246	0.05983	0.05985	0.05979
	RMSE	0.07033	0.07201	<b>0.06488</b>	0.06791	0.07683	0.07682	0.07615
	Tiempo	<b>11.98s</b>	21.85s	147.11s	57.08s	82.49s	70.38s	110.2s

debido a la cantidad de tiempo que requieren. En la Tabla 5.3 se puede observar el modelo obtenido por ajuste a prueba y error, y su versión optimizada por AG con inicialización de RS. El resultado obtenido por el AG fue una mejora del 8.1 % aproximadamente.

**Tabla 5.3:** Comparación resultado de mejor resultado obtenido durante experimentos y su optimización por AG

Modelo/Estrategia	MSE	MAE	RMSE	Tiempo
LSTM/MIMO	0.00370	0.04329	0.06084	1 minuto, 24 segundos
LSTM/MIMO AG	0.00342	0.04001	0.05848	8 días, 3 horas, 8 minutos, 49 segundos

En la Figura 5.1 se puede observar un pronóstico para 24 horas obtenido por el mejor modelo durante los experimentos realizados. Los resultados mostrados muestran el valor real (línea azul) y los valores pronosticados de la velocidad del viento (línea punteada roja). La diferencia de distancia que existe entre la posición que conforma cada uno de los puntos de la gráfica es lo que se conoce como el error, por lo que entre mas cercana es la distancia y existe menor diferencia entre las líneas se considera mejor.



**Figura 5.1:** Gráfica de pronóstico de 24 horas para la velocidad del viento para la serie de tiempo de Aristeo Mercado utilizando LSTM con estrategia MIMO

## 5.2. Resultados de Cointzio

Los resultados obtenidos para la serie de tiempo de Cointzio (ver Tabla 5.4) mostró mejores resultados utilizando el modelo LSTM, particularmente LSTM utilizando estrategia MIMO mostró la mejores resultados en comparación de las otras combinaciones de modelo y estrategias. La segunda combinación que obtuvo un resultado similar fue MLP en conjunto con la estrategia Directa, sin embargo el tiempo de entrenamiento requerido por la estrategia Directa es casi 4 veces mayor. La selección del mejor modelo de los resultados mostrados por la Tabla 5.4 se realizo por medio de la métrica MSE, donde el menor valor representa un menor error. Cabe mencionar que error calculado por el MSE y MAE no están proporcionalmente relacionados, y mientras un conjunto de modelo y estrategia pueden mostrar mejores resultados con el MSE, el MAE puede mostrar mejores resultados. Esto se puede apreciar en los resultados de la Tabla 5.4 para el conjunto de modelo LSTM-MLP y estrategia DirREC.

La diferencia que se presenta entre los resultados donde se obtiene mejor resultado de MAE a pesar de que el modelo posee mejor MSE se debe a que durante el entrenamiento la métrica utilizada para realizar los ajustes fue el MSE.

En la Tabla 5.5 se muestra la optimización para el mejor resultado obtenido por el modelo LSTM con estrategia MIMO, donde su resultado obtiene alrededor de 1.65 % de mejora. Sin embargo, optimizar por medio de AG consume una gran cantidad de tiempo al requerir

**Tabla 5.4:** Resultados obtenidos con los distintos modelos y estrategias utilizando la serie de tiempo de Cointzio

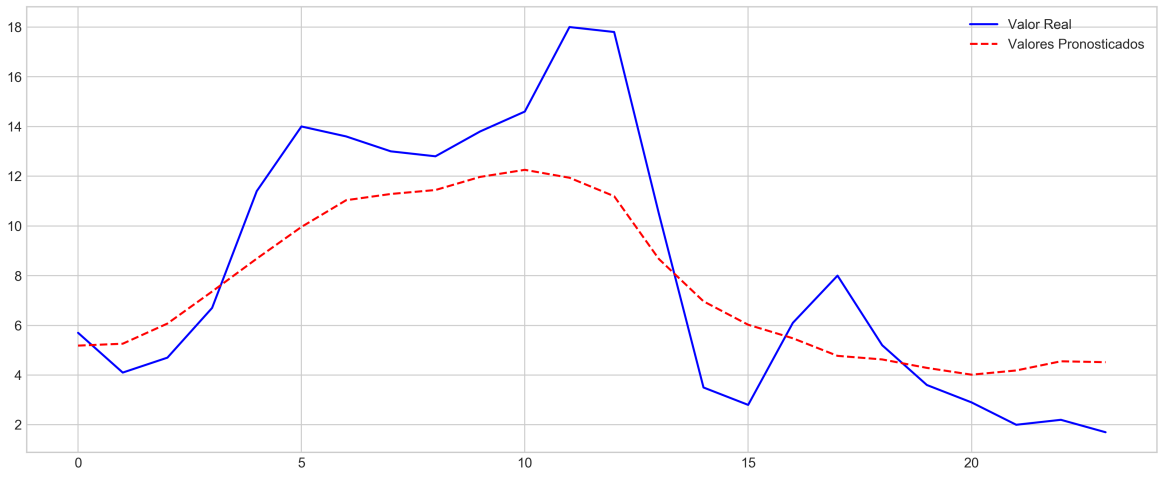
Modelo		k-NN	MLP	LSTM	LSTM-MLP	CNN	CNN-LSTM	CNN-LSTM-MLP
Estrategia de pronóstico	Métrica							
Estrategia Recursiva	MSE	0.01378	0.01071	<b>0.01070</b>	0.01430	0.01471	0.01466	0.01484
	MAE	0.07956	<b>0.07410</b>	0.07434	0.08553	0.08635	0.08858	0.08717
	RMSE	0.11737	0.10348	<b>0.10342</b>	0.11956	0.12128	0.12108	0.12184
	Tiempo	<b>0.03s</b>	78.39s	97.26s	69.82s	27.21s	125.39s	112.69s
Estrategia Directa	MSE	0.01221	<b>0.00996</b>	0.01014	0.01019	0.01341	0.01976	0.01836
	MAE	0.08049	0.07009	0.06949	<b>0.06911</b>	0.08296	0.11047	0.10507
	RMSE	0.11051	<b>0.09982</b>	0.10068	0.10094	0.11579	0.14058	0.13550
	Tiempo	<b>28.89s</b>	1822.26s	589.8s	652.74s	971.9s	571.36s	533.11s
Estrategia MIMO	MSE	0.01221	0.0098	<b>0.00973</b>	0.00975	0.01361	0.01408	0.01429
	MAE	0.08049	0.06933	<b>0.06801</b>	0.06847	0.08450	0.08656	0.08770
	RMSE	0.11051	0.09902	<b>0.09865</b>	0.09873	0.11667	0.11868	0.11956
	Tiempo	<b>0.03s</b>	198.1s	419.27s	456.52s	70.17s	161.14s	102.63s
Estrategia DIRMO	MSE	0.01221	0.00977	<b>0.00976</b>	0.00999	0.01351	0.01492	0.01478
	MAE	0.08	0.06905	<b>0.06790</b>	0.06847	0.08308	0.08704	0.08645
	RMSE	0.11	0.09886	<b>0.09877</b>	0.09996	0.11625	0.12216	0.12156
	Tiempo	<b>5.09s</b>	478.44s	1471.2s	297.68s	252.49s	415.37s	205.59s
Estrategia DirREC	MSE	0.01221	0.01087	<b>0.01013</b>	0.01018	0.01585	0.01965	0.01870
	MAE	0.08049	0.07411	0.06943	<b>0.06905</b>	0.09395	0.11013	0.10632
	RMSE	0.11051	0.10424	<b>0.10063</b>	0.10089	0.12589	0.14019	0.13674
	Tiempo	<b>29.67s</b>	1277.75s	814.3s	857.14s	1101.81s	536.28s	787.04s

entrenar una gran cantidad de modelos, por lo que es importante considerar los modelos y estrategias que serán optimizados.

**Tabla 5.5:** Comparación resultado de mejor resultado obtenido durante experimentos y su optimización por AG

Modelo/Estrategia	MSE	MAE	RMSE	Tiempo
LSTM/MIMO	0.00973	0.06801	0.09865	6 minutos, 59 segundos
LSTM/MIMO AG	0.00957	0.06825	0.09782	9 días, 13 horas, 39 minutos, 15 segundos

En la Figura 5.2 se muestra la gráfica para un pronóstico de 24 horas realizado con el mejor modelo obtenido para la serie de tiempo de Cointzio. Los valores reales (línea azul), representan los valores que son originales a la serie de tiempo utilizados para realizar la evaluación del modelo, mientras que los valores pronosticados (línea roja) son los valores obtenidos por el modelo pronosticador. La diferencia de distancia entre las líneas mostradas por la gráfica de la Figura 5.2 representa el error existente para el pronóstico realizado, donde la diferencia de distancia entre estas líneas es utilizado para calcular el MSE.



**Figura 5.2:** Gráfica de pronóstico de 24 horas para la velocidad del viento para la serie de tiempo de Coitazio utilizando LSTM con estrategia MIMO

### 5.3. Resultados Corrales

Los resultados obtenidos para la serie de tiempo de Corrales (ver Tabla 5.6) muestran mejores resultados con el modelo LSTM utilizando estrategia MIMO en comparación de los demás modelos y estrategias. No obstante, es importante notar que los resultados obtenidos muestran que las demás combinaciones obtienen resultados muy cercanos con menores tiempos de entrenamiento. Es por esto que es importante no considerar solamente la aptitud del modelo, ya que el tiempo que este puede tardar para entrenarse es de gran importancia al reflejar un aumento proporcional al tiempo requerido para optimizar este modelo por medio de AG con inicialización de RS. La selección del mejor modelo de los resultados mostrados por la Tabla 5.6 fue realizada por medio de la métrica MSE, donde el conjunto de modelo y estrategia con el menor número posee el menor error. Sin embargo, existen situación donde el MAE es menor para un modelo distinto donde este posee el menor MSE. Esto se debe a que el MAE no es directamente proporcional al MSE y para este trabajo los modelos fueron ajustados utilizando el MSE.

En la Tabla 5.7 se muestran los resultados obtenidos por el modelo con mejor aptitud y su versión optimizada por AG con inicialización por RS. Se obtuvo una mejora del 5% en basado en el MSE para un lapso de mas de 6 días de ajuste automático.

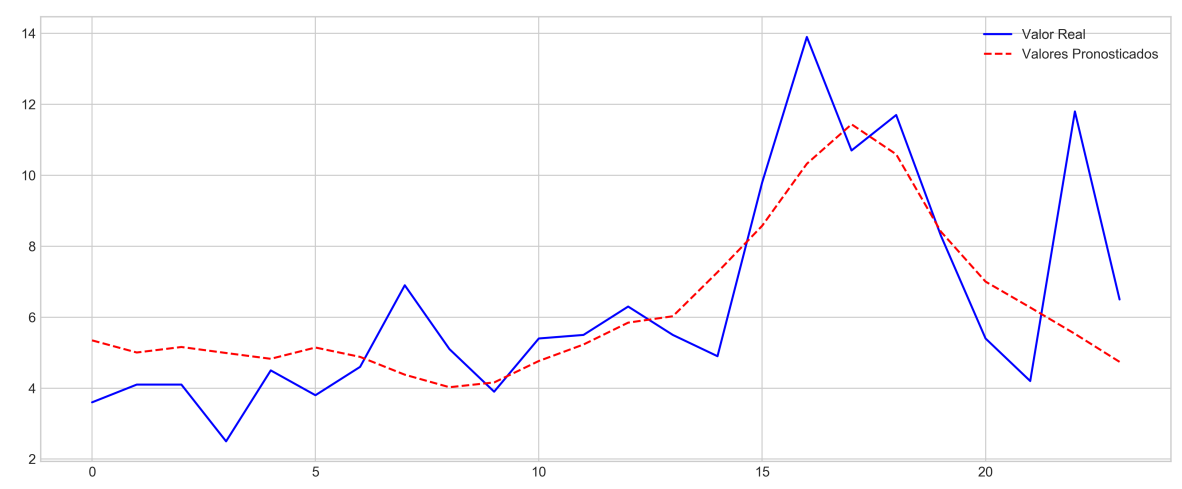
**Tabla 5.6:** Resultados obtenidos con los distintos modelos y estrategias utilizando la serie de tiempo de Corrales

Modelo		k-NN	MLP	LSTM	LSTM-MLP	CNN	CNN-LSTM	CNN-LSTM-MLP
Estrategia de pronóstico	Métrica							
Estrategia Recursiva	MSE	0.00417	<b>0.00354</b>	0.00372	0.00384	0.00483	0.00490	0.00511
	MAE	0.04404	<b>0.04359</b>	0.04626	0.04698	0.05264	0.05337	0.05505
	RMSE	0.06459	<b>0.05949</b>	0.06097	0.06194	0.06950	0.06999	0.07146
	Tiempo	<b>0.011s</b>	28.47s	438.74s	249.5s	31.76s	24.47s	57.63s
Estrategia Directa	MSE	0.00412	<b>0.00345</b>	<b>0.00345</b>	0.00354	0.00457	0.00471	0.00462
	MAE	0.04779	<b>0.04259</b>	0.04264	0.04316	0.05036	0.05216	0.05171
	RMSE	0.06419	0.05877	<b>0.05871</b>	0.05948	0.06761	0.06862	0.06795
	Tiempo	<b>11.30s</b>	328.36s	708.89s	693.44s	912.34s	800.32s	521.83s
Estrategia MIMO	MSE	0.00412	0.00350	<b>0.00339</b>	0.00351	0.00444	0.00489	0.00455
	MAE	0.04779	0.04337	<b>0.04195</b>	0.04312	0.04971	0.05398	0.05082
	RMSE	0.06419	0.05913	<b>0.05823</b>	0.05926	0.06665	0.06993	0.06748
	Tiempo	<b>0.013s</b>	134.57s	1210.93s	307.29s	79.2s	154.63s	51.66s
Estrategia DIRMO	MSE	0.00412	0.00359	<b>0.00351</b>	0.00356	0.00449	0.00462	0.00454
	MAE	0.04779	0.04409	<b>0.04323</b>	0.04352	0.05019	0.05139	0.05063
	RMSE	0.06419	0.05989	<b>0.05923</b>	0.05968	0.06698	0.06795	0.06741
	Tiempo	<b>1.99s</b>	182.93s	621.79s	236.45s	263.72s	402.30	217.06s
Estrategia DirREC	MSE	0.00373	0.00364	<b>0.00345</b>	0.00354	0.00485	0.00462	0.00468
	MAE	0.04489	0.04400	<b>0.04262</b>	0.04318	0.05331	0.05149	0.05235
	RMSE	0.06109	0.06032	<b>0.05869</b>	0.05949	0.06965	0.06794	0.06845
	Tiempo	<b>12.41s</b>	548.59s	628.5	974.31s	818.69s	1431.13	840s

**Tabla 5.7:** Comparación resultado de mejor resultado obtenido durante experimentos y su optimización por AG

Modelo/Estrategia	MSE	MAE	RMSE	Tiempo
LSTM/MIMO	0.00339	0.04195	0.05823	20 minutos, 11 segundos
LSTM/MIMO AG	0.003218	0.04004	0.05673	6 días, 23 horas, 21 minutos, 2 segundos

En la Figura 5.3 se muestra la gráfica para un pronóstico de 24 horas entregado por este modelo.



**Figura 5.3:** Gráfica de pronóstico de 24 horas para la velocidad del viento para la serie de tiempo de Corrales utilizando LSTM con estrategia MIMO

## 5.4. Resultados La Piedad

Los resultados obtenidos por la serie de tiempo de La Piedad (ver Tabla 5.8) muestran que los modelos utilizando estrategia DIRMO, particularmente la combinación del modelo LSTM y MLP, superan los otros conjuntos de modelo y estrategias. Este combinación de modelo y estrategia permite una excelente flexibilidad de ajuste, por lo que su potencial de mejora es grande. La selección del mejor conjunto de modelo y estrategia mostrados por la Tabla 5.8 se realizo por medio de la métrica MSE, donde un menor número representa un menor error.

**Tabla 5.8:** Resultados obtenidos con los distintos modelos y estrategias utilizando la serie de tiempo de La Piedad

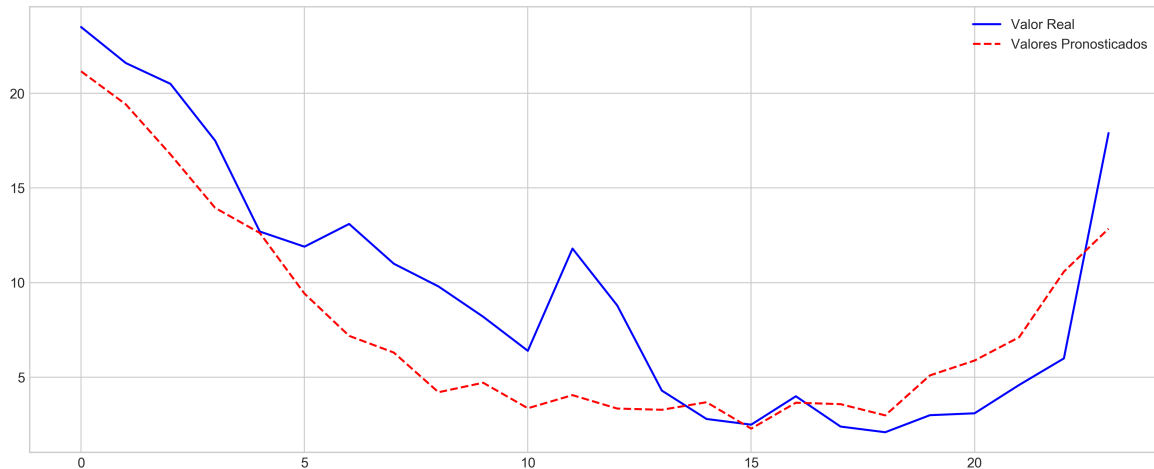
Modelo		k-NN	MLP	LSTM	LSTM-MLP	CNN	CNN-LSTM	CNN-LSTM-MLP
Estrategia de pronóstico	Métrica							
Estrategia Recursiva	MSE	0.02328	0.01763	<b>0.01628</b>	0.01683	0.01989	0.02070	0.02443
	MAE	0.11961	0.10142	<b>0.09673</b>	0.10024	0.10613	0.10644	0.11875
	RMSE	0.15258	0.13278	<b>0.12761</b>	0.12972	0.14102	0.14389	0.15630
	Tiempo	<b>0.021s</b>	37.67s	71.41s	75.39s	50.99s	87.05s	63.73s
Estrategia Directa	MSE	0.02037	<b>0.01606</b>	0.01684	0.01680	0.02139	0.02109	0.02130
	MAE	0.11048	<b>0.09377</b>	0.09680	0.09680	0.10859	0.10662	0.10738
	RMSE	0.14273	<b>0.12674</b>	0.12976	0.12961	0.14627	0.14523	0.14595
	Tiempo	<b>15.28s</b>	134.69s	1910.41s	833.98s	1108.24s	791.5s	479.69s
Estrategia MIMO	MSE	0.02037	0.01648	0.01589	<b>0.01515</b>	0.01998	0.02352	0.02063
	MAE	0.11048	0.09611	0.09368	<b>0.093029</b>	0.10650	0.11793	0.10857
	RMSE	0.14273	0.12837	0.12606	<b>0.12308</b>	0.14134	0.15338	0.14365
	Tiempo	<b>0.019s</b>	11.8s	64.66s	670.3s	150.28s	401.45s	87.43s
Estrategia DIRMO	MSE	0.02037	0.01641	<b>0.01595</b>	0.01643	0.02050	0.02097	0.02184
	MAE	0.11048	0.09549	<b>0.09284</b>	0.09622	0.10674	0.10786	0.11115
	RMSE	0.14273	0.12808	<b>0.12630</b>	0.12816	0.14318	0.14482	0.14780
	Tiempo	<b>2.75s</b>	54.9s	302.69s	191.51s	388.49s	423.11s	240.24s
Estrategia DirREC	MSE	0.02011	0.01684	<b>0.01534</b>	0.01677	0.02239	0.02123	0.02069
	MAE	0.10971	0.09740	<b>0.09073</b>	0.09671	0.11742	0.10676	0.10610
	RMSE	0.14182	0.12976	<b>0.12386</b>	0.12948	0.14962	0.14570	0.14384
	Tiempo	<b>18.46s</b>	195s	1175.64s	1589.10s	1000.51s	1138.44s	792.92s

En la Tabla 5.9 se muestra el modelo y estrategia con mejor aptitud entre los resultados obtenidos, y su versión optimizada por AG con inicialización por RS. El porcentaje de mejora obtenido por el AG fue de alrededor del 17.72 %.

**Tabla 5.9:** Comparación resultado de mejor resultado obtenido durante experimentos y su optimización por AG

Modelo/Estrategia	MSE	MAE	RMSE	Tiempo
LSTM-MLP/DIRMO	0.01515	0.093029	0.12308	11 minutos, 10 segundos
LSTM-MLP/DIRMO AG	0.01245	0.08217	0.11157	12 días, 7 horas, 58 minutos, 27 segundos

En la Figura 5.4 se muestra un pronóstico de 24 horas obtenido por este modelo. El valor



**Figura 5.4:** Gráfica de pronóstico de 24 horas para la velocidad del viento para la serie de tiempo de La Piedad utilizando LSTM-MLP con estrategia DIRMO

real(línea azul) representa los valores reales utilizados para validar el modelo, donde el error es calculado utilizando la métrica MSE por medio de la diferencia que existe entre los valores reales y los valores pronosticados(línea punteada roja).

## 5.5. Resultados Melchor Campo

Para la serie de tiempo de Melchor Campo los resultados obtenidos (ver Tabla 5.10) muestra que los modelos que utilizaron estrategias MIMO y DIRMO fueron las que obtuvieron un menor error (basado en el MSE). El modelo que obtuvo la mejor aptitud fue un LSTM utilizando la estrategia DIRMO seguido por LSTM-MLP con MIMO. El modelo con mejor aptitud presenta una gran flexibilidad y facilidad de obtener mejores resultados, esto debido a la estructura que presenta la estrategia DIRMO. La selección del mejor conjunto de modelo y estrategia entre los resultados mostrados por la Tabla 5.10 se realizó por medio de la métrica MSE, donde el menor número representa un menor error. Similarmente a casos anteriores, los modelos pueden presentar un mejor resultado por la métrica MAE debido a que no es directamente proporcional a la métrica MSE. Esto debido a que los modelos obtenidos fueron ajustados por medio del MSE. Sin embargo existen ocasiones en las que a pesar de que el modelo fue ajustado utilizando una métrica, algunos de los modelos pueden mostrar mejores resultados con una métrica distinta. Este caso se presenta en los resultados de la Tabla 5.10

para el conjunto de modelo LSTM y estrategia DirREC.

**Tabla 5.10:** Resultados obtenidos con los distintos modelos y estrategias utilizando la serie de tiempo de Melchor Campo

Modelo		k-NN	MLP	LSTM	LSTM-MLP	CNN	CNN-LSTM	CNN-LSTM-MLP
Estrategia de pronóstico	Métrica							
Estrategia Recursiva	MSE	0.01743	0.01867	0.02048	<b>0.01517</b>	0.01686	0.01970	0.01919
	MAE	0.09801	0.09359	0.09515	<b>0.09040</b>	0.09675	0.10493	0.10659
	RMSE	0.13203	0.13663	0.14313	<b>0.12317</b>	0.12986	0.14034	0.13852
	Tiempo	<b>0.044s</b>	41.3s	450.83s	74.27s	51.67s	57.02s	50.81s
Estrategia Directa	MSE	0.01607	<b>0.01438</b>	0.01478	0.01453	0.01647	0.02017	0.01972
	MAE	0.09614	<b>0.08941</b>	0.08990	0.09049	0.09632	0.10487	0.10367
	RMSE	0.12676	<b>0.11992</b>	0.12159	0.12054	0.12832	0.14203	0.14044
	Tiempo	<b>22.54s</b>	318.6s	477.7s	662.28s	1166.89s	531.35s	509.5s
Estrategia MIMO	MSE	0.12676	0.01432	0.01414	<b>0.01412</b>	0.01641	0.01868	0.01929
	MAE	0.01607	0.08911	0.08788	<b>0.08864</b>	0.09590	0.10354	0.10460
	RMSE	0.09614	0.11967	0.11891	<b>0.11882</b>	0.12809	0.13669	0.13887
	Tiempo	<b>0.018s</b>	51.82s	451.55s	364.98s	86.51s	73.56s	84.92s
Estrategia DIRMO	MSE	0.01607	0.01430	<b>0.01399</b>	0.01463	0.01672	0.01812	0.01856
	MAE	0.09614	0.08880	<b>0.08739</b>	0.09026	0.09713	0.10236	0.10329
	RMSE	0.12676	0.11958	<b>0.11829</b>	0.12097	0.12930	0.13461	0.13622
	Tiempo	<b>3.96s</b>	197.46s	2244.96s	179.08s	318.68s	389.34s	317.33s
Estrategia DirREC	MSE	0.01569	0.01471	0.01478	<b>0.01452</b>	0.01795	0.01990	0.02002
	MAE	0.09403	0.09010	<b>0.08982</b>	0.09048	0.10119	0.10451	0.10423
	RMSE	0.12528	0.12128	0.12157	<b>0.12063</b>	0.13398	0.14105	0.14150
	Tiempo	<b>26.10s</b>	420.48s	688.19s	901.98s	1500.34s	844.66s	1230.82s

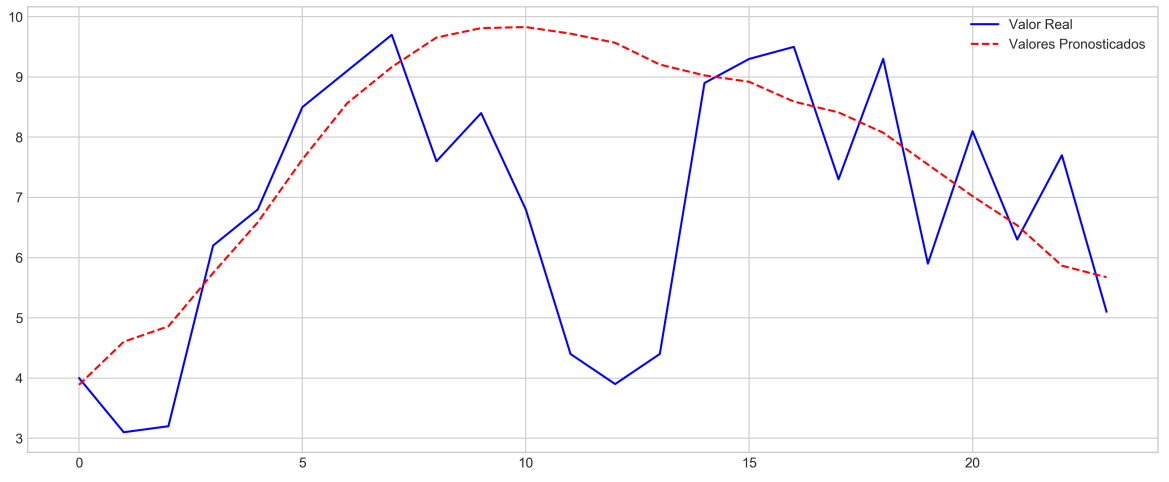
En la Tabla 5.11 se muestran el modelo y estrategia con mejor aptitud obtenidos durante los experimentos y su versión optimizada por AG con inicialización por RS, el cual muestra un porcentaje de mejora fue de alrededor del 7 %. Entre los modelos optimizados, éste fue el que consumió más tiempo debido a la velocidad que se puede entrenar un modelo con estrategia DIRMO.

**Tabla 5.11:** Comparación resultado de mejor resultado obtenido durante experimentos y su optimización por AG

Modelo/Estrategia	MSE	MAE	RMSE	Tiempo
LSTM/DIRMO	0.01399	0.08739	0.11829	37 minutos 24 segundos
LSTM/DIRMO AG	0.01301	0.08174	0.11406	9 días, 14 horas, 34 minutos, 47 segundos

La Figura 5.5 se muestra un pronóstico de 24 horas utilizando este modelo y estrategia. Los valores reales (línea azul) representan los valores utilizados para realizar la evaluación del modelo. Estos valores son utilizados para calcular el error por medio de la métrica MSE para el modelo, donde se utiliza la diferencia de distancia entre los valores reales y los valores pronosticados (línea roja punteada) mostrados por la gráfica de la Figura 5.5.





**Figura 5.5:** Gráfica de pronóstico de 24 horas para la velocidad del viento para la serie de tiempo de Melchor Campo utilizando LSTM con estrategia DIRMO

---

# Capítulo 6

## Conclusiones

---

### 6.1. Conclusiones

El uso de pronosticadores juega un papel importante en la industria, especialmente para el campo relacionado a la generación de energía eléctrica por medios eólicos. Los pronosticadores de la velocidad del viento ya tiene décadas en uso por medio de técnicas estadísticas, sin embargo, las tecnologías de Aprendizaje Maquina (ML) se ha popularizado en años recientes. A pesar de los excelentes resultados que han entregado estos modelos para el pronóstico de la velocidad del viento, estos deben ser reforzados con estrategias de pronóstico a múltiples pasos para obtener predicciones de la velocidad del viento a mayores horizontes.

Es por lo anterior que durante el desarrollo de este trabajo se realizaron experimentos y análisis de 7 modelos de ML en conjunto con 5 estrategias de pronóstico a múltiples pasos mas conocidas. Esto con el objetivo de obtener mejores pronósticos para 24 horas adelante de la velocidad del viento.

Durante el análisis de los resultados se observó que el modelo LSTM y estrategia MIMO que obtuvieron los mejores experimentos entre las distintas combinaciones utilizadas para los experimentos de este trabajo. Sin embargo, esto puede variar dependiendo la serie de tiempo con la cual se esta trabajando, ya que para algunas de las series de tiempo los mejores resultados se obtuvieron empleando la estrategia DIRMO y Recursiva. Otro punto importante es considerar el tiempo y poder computacional requeridos para obtener un modelo, ya que esto puede afectar el costo final del mismo y posible implementación.

A pesar no mostrar los mejores resultados con los modelos basados en CNN, estos poseen

la mayor cantidad de hiperparámetros ajustables, por lo que el poder computacional limitado durante la ejecución de los experimentos realizados durante este trabajo pueden haber impactado de forma negativa sus resultados.

Otro factor observado durante el desarrollo de este trabajo fue la importancia que juegan los datos (series de tiempo) utilizados para entrenar los modelos. Una serie de tiempo específica puede obtener mejores resultados con un modelo, mientras que el mismo modelo puede obtener pésimos resultados con otra de las series de tiempo. Es por esto que es conveniente considerar múltiples modelos para la búsqueda de pronósticos con buena aptitud. Además, es fundamental considerar las técnicas de preprocesamiento utilizadas para buscar reducir el ruido y eliminar los datos atípicos para reducir el error obtenido por los modelos pronosticadores.

Las técnicas de IA demostraron ser capaces de obtener resultados favorables al realizar pronósticos de la velocidad del viento; sin embargo, estas requieren una gran cantidad de ajuste de hiperparámetros para obtener dichos resultados. Además, la calidad de los datos tiene un gran impacto en el desempeño obtenido por el modelo pronosticador, especialmente los datos atípicos. Dicho lo anterior, existen muchas técnicas para el pronóstico de series de tiempo que pueden ser implementadas a la metodología propuesta en este trabajo o bien contribuir a otros.

## **6.2. Aportaciones**

Este trabajo aporta una metodología para la implementación de modelos de ML en conjunto con estrategias de pronóstico a múltiples pasos en el tiempo. Esta metodología incluye desde el procesamiento y preparación de datos, implementaciones de modelo y estrategias y la optimización de estos modelos por medio de Algoritmos Genéticos (AG) inicializados por una Búsqueda Aleatoria (RS).

Esta metodología no solamente es posible implementarla en pronósticos de la velocidad del viento, ya que es aplicable en cualquier problema que requiera realizar pronósticos de cualquier variable de interés.

### **6.3. Trabajo Futuro**

Como trabajo futuro se sugiere el uso de técnicas para el preprocesamiento de los datos que ayuden a mejorar la aptitud de los modelos pronosticadores, tales como la descomposición de señal por la transformada de Wavelet (H. Liu, Tian & Li, 2015).

Además se buscaran técnicas de optimización de hiperparámetros que lleven a resultados similares en un menor tiempo y ampliar la cantidad de parámetros y tamaño de estructura de los modelos de ML utilizados durante este trabajo, con el objetivo de tener un panorama mas amplio de posible mejora de resultados.

# Bibliografía

---

- Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... Isard, M. et al. (2016). Tensorflow: A system for large-scale machine learning. En *12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16)* (pp. 265-283).
- Abas, N., Kalair, A. & Khan, N. (2015). Review of fossil fuels and future energy technologies. *Futures*, 69, 31-49.
- An, N. H. & Anh, D. T. (2015). Comparison of strategies for multi-step-ahead prediction of time series using neural network. En *2015 International Conference on Advanced Computing and Applications (ACOMP)* (pp. 142-149). IEEE.
- Beasley, D., Bull, D. R. & Martin, R. R. (1993). An overview of genetic algorithms: Part 1, fundamentals. *University computing*, 15(2), 56-69.
- Bergstra, J. S., Bardenet, R., Bengio, Y. & Kégl, B. (2011). Algorithms for hyper-parameter optimization. En *Advances in neural information processing systems* (pp. 2546-2554).
- Bhaskar, M., Jain, A. & Srinath, N. V. (2010). Wind speed forecasting: Present status. En *2010 International Conference on Power System Technology* (pp. 1-6). IEEE.

- Bhattacharya, M., Paramati, S. R., Ozturk, I. & Bhattacharya, S. (2016). The effect of renewable energy consumption on economic growth: Evidence from top 38 countries. *Applied Energy*, 162, 733-741.
- Boettiger, C. (2015). An introduction to Docker for reproducible research. *ACM SIGOPS Operating Systems Review*, 49(1), 71-79.
- Bontempi, G., Taieb, S. B. & Le Borgne, Y.-A. (2012). Machine learning strategies for time series forecasting. En *European business intelligence summer school* (pp. 62-77). Springer.
- Bottiau, J., Vallée, F., De Grève, Z. & Toubeau, J.-F. (2018). Leveraging provision of frequency regulation services from wind generation by improving day-ahead predictions using LSTM neural networks. En *2018 IEEE International Energy Conference (ENERGY-CON)* (pp. 1-6). IEEE.
- Boyes, W. (2009). *Instrumentation reference book*. Butterworth-Heinemann.
- Brockwell, P. J., Davis, R. A. & Calder, M. V. (2002). *Introduction to time series and forecasting*. Springer.
- Chang, W.-Y. (2014). A literature review of wind forecasting methods. *Journal of Power and Energy Engineering*, 2(4).
- Chen, X.-W. & Lin, X. (2014). Big data deep learning: challenges and perspectives. *IEEE access*, 2, 514-525.
- Dalto, M., Matuško, J. & Vašak, M. (2015). Deep neural networks for ultra-short-term wind forecasting. En *2015 IEEE International Conference on Industrial Technology (ICIT)* (pp. 1657-1663). IEEE.
- Dell, R. & Rand, D. A. J. (2004). *Clean energy*. Royal Society of Chemistry.

- Eckhart, M., El-Ashry, M., Hales, D., Hamilton, K., Rae, P. & Bariloche, F. (2018). Renewables 2018 Global Status Report.
- Erdem, E. & Shi, J. (2011). ARMA based approaches for forecasting the tuple of wind speed and direction. *Applied Energy*, 88(4), 1405-1414.
- Ewing, R. C. (2015). Long-term storage of spent nuclear fuel. *Nature Materials*, 14(3), 252.
- Feurer, M. & Hutter, F. (2019). Hyperparameter optimization. En *Automated Machine Learning* (pp. 3-33). Springer.
- Flores, J. J., Graff, M. & Rodriguez, H. (2012). Evolutive design of ARMA and ANN models for time series forecasting. *Renewable Energy*, 44, 225-230.
- Fortin, F.-A., De Rainville, F.-M., Gardner, M.-A., Parizeau, M. & Gagné, C. (2012). DEAP: Evolutionary Algorithms Made Easy. *Journal of Machine Learning Research*, 13, 2171-2175.
- Goodfellow, I., Bengio, Y. & Courville, A. (2016). *Deep learning*. MIT press.
- Gulli, A. & Pal, S. (2017). *Deep Learning with Keras*. Packt Publishing Ltd.
- Hagan, M. T., Demuth, H. B., Beale, M. H. & De Jesús, O. (1996). *Neural network design*. Pws Pub. Boston.
- Harvey, D. I., Leybourne, S. J. & Newbold, P. (1998). Tests for forecast encompassing. *Journal of Business & Economic Statistics*, 16(2), 254-259.
- Harvey, D. & Newbold, P. (2000). Tests for multiple forecast encompassing. *Journal of Applied Econometrics*, 15(5), 471-482.
- Hyndman, R. J. & Koehler, A. B. (2006). Another look at measures of forecast accuracy. *International journal of forecasting*, 22(4), 679-688.
- Jain, A. K., Dubes, R. C. et al. (1988). *Algorithms for clustering data*. Prentice hall Englewood Cliffs.

- Kavasseri, R. G. & Seetharaman, K. (2009). Day-ahead wind speed forecasting using f-ARIMA models. *Renewable Energy*, 34(5), 1388-1393.
- Kersting, K. (2018). Machine Learning and Artificial Intelligence: Two Fellow Travelers on the Quest for Intelligent Behaviour in Machines. *Frontiers in Big Data*, 1, 6.
- Kline, D. M. (2004). Methods for multi-step time series forecasting neural networks. En *Neural networks in business forecasting* (pp. 226-250). IGI Global.
- Kumar, Y., Ringenberg, J., Depuru, S. S., Devabhaktuni, V. K., Lee, J. W., Nikolaidis, E., ... Afjeh, A. (2016). Wind energy: Trends and enabling technologies. *Renewable and Sustainable Energy Reviews*, 53, 209-224.
- LeCun, Y., Bengio, Y. & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436.
- Lei, M., Shiyan, L., Chuanwen, J., Hongling, L. & Yan, Z. (2009). A review on the forecasting of wind speed and generated power. *Renewable and Sustainable Energy Reviews*, 13(4), 915-920.
- Lin, H. W., Tegmark, M. & Rolnick, D. (2017). Why does deep and cheap learning work so well? *Journal of Statistical Physics*, 168(6), 1223-1247.
- Liu, D., Niu, D., Wang, H. & Fan, L. (2014). Short-term wind speed forecasting using wavelet transform and support vector machines optimized by genetic algorithm. *Renewable Energy*, 62, 592-597.
- Liu, H., Tian, H.-q. & Li, Y.-f. (2015). Four wind speed multi-step forecasting models using extreme learning machines and signal decomposing algorithms. *Energy conversion and management*, 100, 16-22.



- Liu, H., Tian, H.-q., Pan, D.-f. & Li, Y.-f. (2013). Forecasting models for wind speed using wavelet, wavelet packet, time series and Artificial Neural Networks. *Applied Energy*, 107, 191-208.
- Manwell, J. F., McGowan, J. G. & Rogers, A. L. (2010). *Wind energy explained: theory, design and application*. John Wiley & Sons.
- Mao, Y. & Shaoshuai, W. (2016). A review of wind power forecasting & prediction. En *2016 International Conference on Probabilistic Methods Applied to Power Systems (PMAPS)* (pp. 1-7). IEEE.
- McCarthy, J. (2004). What is Artificial Intelligence?
- Michael, N. (2005). *Artificial intelligence a guide to intelligent systems*. Addison Wesley.
- Mitchell, T. M. (1997). *Machine Learning*. McGraw-Hill Science/Engineering/Math.
- Montgomery, D. C., Jennings, C. L. & Kulahci, M. (2015). *Introduction to time series analysis and forecasting*. John Wiley & Sons.
- Nayak, S., Misra, B. & Behera, H. (2014). Impact of data normalization on stock index forecasting. *Int. J. Comp. Inf. Syst. Ind. Manag. Appl*, 6, 357-369.
- Nielsen, M. A. (2015). *Neural networks and deep learning*. Determination press San Francisco, CA, USA:
- Oh, J., Guo, X., Lee, H., Lewis, R. L. & Singh, S. (2015). Action-conditional video prediction using deep networks in atari games. En *Advances in neural information processing systems* (pp. 2863-2871).
- Oliphant, T. E. (2007). Python for scientific computing. *Computing in Science & Engineering*, 9(3), 10-20.

- Pasero, E., Raimondo, G. & Ruffa, S. (2010). MULP: a multi-layer perceptron application to long-term, out-of-sample time series prediction. En *International Symposium on Neural Networks* (pp. 566-575). Springer.
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Dubourg, V. et al. (2011). Scikit-learn: Machine learning in Python. *Journal of machine learning research*, 12(Oct), 2825-2830.
- Potter, C. W. & Negnevitsky, M. (2006). Very short-term wind forecasting for Tasmanian power generation. *IEEE Transactions on Power Systems*, 21(2), 965-972.
- Quaschnig, V. V. (2019). *Renewable energy and climate change*. Wiley.
- Rangel, H. R., Puig, V., Farias, R. L. & Flores, J. J. (2017). Short-term demand forecast using a bank of neural network models trained using genetic algorithms for the optimal management of drinking water networks. *Journal of Hydroinformatics*, 19(1), 1-16.
- Rodríguez, Jessica, Arcos, Álvarez et al. (2017). Reporte de Avances de Energías Limpias, Al Primer Semestre de 2017. *SENER. Presentation for the "Consejo Consultivo para la Transición Energética."* December, 11, 2017.
- Saad, E. W., Prokhorov, D. V. & Wunsch, D. C. (1998). Comparative study of stock trend prediction using time delay, recurrent and probabilistic neural networks. *IEEE Transactions on neural networks*, 9(6), 1456-1470.
- Salza, P., Ferrucci, F. & Sarro, F. (2016). Develop, deploy and execute parallel genetic algorithms in the cloud. En *Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion* (pp. 121-122). ACM.

- Santamaría-Bonfil, G., Reyes-Ballesteros, A. & Gershenson, C. (2016). Wind speed forecasting for wind farms: A method based on support vector regression. *Renewable Energy*, 85, 790-809.
- Saroha, S. & Aggarwal, S. (2014). Multi step ahead forecasting of wind power by genetic algorithm based neural networks. En *2014 6th IEEE Power India International Conference (PIICON)* (pp. 1-6). IEEE.
- Shcherbakov, M. V., Brebels, A., Shcherbakova, N. L., Tyukov, A. P., Janovsky, T. A. & Kamaev, V. A. (2013). A survey of forecast error measures. *World Applied Sciences Journal*, 24(24), 171-176.
- Smola, A. & Vishwanathan, S. (2008). *Introduction to machine learning*. Cambridge University Press.
- Soman, S. S., Zareipour, H., Malik, O. & Mandal, P. (2010). A review of wind power and wind speed forecasting methods with different time horizons. En *North American Power Symposium 2010* (pp. 1-8). IEEE.
- Sutton, R. S. & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press.
- Taieb, S. B., Bontempi, G., Atiya, A. F. & Sorjamaa, A. (2012). A review and comparison of strategies for multi-step ahead time series forecasting based on the NN5 forecasting competition. *Expert systems with applications*, 39(8), 7067-7083.
- Taieb, S. B., Bontempi, G., Sorjamaa, A. & Lendasse, A. (2009). Long-term prediction of time series by combining direct and mimo strategies. En *2009 International Joint Conference on Neural Networks* (pp. 3054-3061). IEEE.
- Tsay, R. S. (1988). Outliers, level shifts, and variance changes in time series. *Journal of forecasting*, 7(1), 1-20.

- Velo, R., López, P. & Maseda, F. (2014). Wind speed estimation using multilayer perceptron. *Energy Conversion and Management*, 81, 1-9.
- Wang, Z. & Oates, T. (2015). Encoding time series as images for visual inspection and classification using tiled convolutional neural networks. En *Workshops at the Twenty-Ninth AAAI Conference on Artificial Intelligence*.
- Wu, Y.-K. & Hong, J.-S. (2007). A literature review of wind forecasting technology in the world. En *2007 IEEE Lausanne Power Tech* (pp. 504-509). IEEE.
- Xiaoyun, Q., Xiaoning, K., Chao, Z., Shuai, J. & Xiuda, M. (2016). Short-term prediction of wind power based on deep long short-term memory. En *2016 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)* (pp. 1148-1152). IEEE.
- Young, S. R., Rose, D. C., Karnowski, T. P., Lim, S.-H. & Patton, R. M. (2015). Optimizing deep learning hyper-parameters through an evolutionary algorithm. En *Proceedings of the Workshop on Machine Learning in High-Performance Computing Environments* (p. 4). ACM.
- Zhu, A., Li, X., Mo, Z. & Wu, R. (2017). Wind power prediction based on a convolutional neural network. En *2017 International Conference on Circuits, Devices and Systems (ICCDs)* (pp. 131-135). IEEE.