UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

Repositorio Digital USM **https://repositorio.usm.cl**

Tesis USM TESIS de Pregrado de acceso ABIERTO

2020-11

MECANISMOS DE ATENCIÓN EN REDES CONVOLUCIONALES PARA PRONÓSTICO DE POTENCIA EÓLICA

ACEVEDO GONZALEZ, SEBASTIAN ENRIQUE

https://hdl.handle.net/11673/49700

Repositorio Digital USM, UNIVERSIDAD TECNICA FEDERICO SANTA MARIA

UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE

"MECANISMOS DE ATENCIÓN EN REDES CONVOLUCIONALES PARA PRONÓSTICO DE POTENCIA EÓLICA"

SEBASTIÁN ENRIQUE ACEVEDO GONZÁLEZ

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL EN INFORMÁTICA

Profesor Guía: Héctor Allende Olivares
Profesor Correferente: Carlos Valle Vidal

Noviembre - 2020

DEDICATORIA

Dedico esta memoria a mi familia, amigos, novia y seres queridos en general, quienes han estado junto a mí en este año turbulento y difícil.

AGRADECIMIENTOS

Agradezco profundamente a la universidad, quien me ayudo a madurar y crecer. El proceso por el que me hizo pasar me ha cambiado e iluminado, y sin ella quien sabe dónde estaría ahora. Agradezco a mis profesores y compañeros, pues por ellos soy la persona que soy ahora.

Agradezco en particular a los profesores Héctor Allende y Carlos Valle, quienes me guiaron a través del proceso de memoria y titulación. Agradezco también a los miembros del seminario INF-462, quienes me ayudaron a mejorar este trabajo.

RESUMEN

Resumen— Los problemas medioambientales que traen las energías tradicionales han llevado a la búsqueda de fuentes de energía limpias y renovables, entre las que la energía eólica destaca por su baja huella ambiental. Sin embargo, la naturaleza estocástica del viento pone un desafío en la integración de esta fuente de energía en el sistema eléctrico.

En esta memoria se busca implementar un modelo para pronosticar la potencia eólica mediante redes convolucionales con mecanismos de atención, usando series de datos obtenidas de la operación de los parques eólicos Canela y El Totoral. Se compararán los resultados con otros métodos de Deep Learning del estado del arte y el método estadístico ARIMA, mostrando que el modelo propuesto presenta mejores resultados en el pronóstico de 1 a 12 horas.

Palabras Clave—Deep Learning | Mecanismos de Atención | Energía Eólica

Abstract— The environmental problems caused by fossil energies encourage the search for clean and renewable energy sources, among which wind energy stands out for its low environmental footprint. However, the stochastic nature of the wind brings a challenge on the integration of this energy source into the electric system.

This work implements a model to forecast wind power through convolutional neural networks with attention mechanisms, using data series obtained from the operation of Canela and El Totoral wind farms. The results will be compared with other state-of-the-art Deep Learning methods and the ARIMA statistical method, showing that the proposed model presents better results in the one-hour forecast.

Keywords—Deep Learning | Attention Mechanisms | Wind Energy

GLOSARIO

- ACF: Autocorrelation Function
- ANN: Artificial Neural Network
- AR: Autoregresive
- ARCH: Autoregressive Conditional Heteroscedasticity
- ARFIMA: Autoregressive Fractional Integrated Moving Average
- ARIMA: Autoregressive Integrated Moving Average
- ARMA: Autoregresive
- CEN: Coordinador Eléctrico Nacional
- CNN: Convolutional Neural Network
- ERNC: Energía Renovable No Convencional
- GARCH: Generalized Autoregressive Conditional Heteroscedasticity
- GRU: Gated Recurrent Unit
- LSTM: Long Short Term Memory
- MA: Moving Average
- MAE: Mean Absolute Error
- PACF: Partial Autocorrelation Function
- RMSE: Root Mean Squared Error
- RNN: Recurrent Neural Network
- SARIMA: Seasonal Autoregressive Integrated Moving Average.
- SEN: Sistema Eléctrico Nacional

INDICE DE CONTENIDOS

RESUMEN 4

INDICE DE TABLAS 9

INTRODUCCIÓN 11

1. DEFINICIÓN DEL PROBLEMA 12

1.1 Objetivos 15

1.1.1 Objetivo general 15

1.1.2 Objetivos Específicos 15

2. MARCO CONCEPTUAL 16

2.1 Series de tiempo 16

2.2 Modelo ARIMA 17

2.3 Modelo GARCH 19

2.4 Redes Neuronales Feed Forward 21

2.5 Redes Neuronales Convolucionales 24

2.6 Redes Neuronales Recurrentes 26

3. PROPUESTA DE SOLUCION 30

3.1 Caso de estudio 30

3.2 Propuesta 30

3.3 Validación del modelo 32

3.4 Búsqueda de hiperparámetros 34

3.5 Preprocesamiento de los datos 36

3.6 Lenguaje de programación 36

4. VALIDACIÓN DE LA SOLUCIÓN 38

4.1 Análisis exploratorio 38

4.2 Resultados métodos estadísticos 42

4.3 Resultados redes neuronales 44

4.4 Resultados totales 51

5. CONCLUSIONES Y TRABAJO FUTURO 53

5.1 Trabajo futuro 54

REFERENCIAS BIBLIOGRÁFICAS 55

APÉNDICE A: RESULTADOS DE BÚSQUEDA ALEATORIA 59

APÉNDICE B: RESULTADOS DE MODELOS DE MACHINE LEARNING 61

INDICE DE FIGURAS

Ilustración 1: Diagrama de red neuronal feed forward 22

Ilustración 2: Diagrama de convolución 1d 25

Ilustración 3: Esquema de Skip Connection 26

Ilustración 4: Mecanismo de atención en traducción 28

Ilustración 5: Esquema de datos de entrada y salida 30

Ilustración 6: Diagrama de propuesta de arquitectura CNN-Attention 32

Ilustración 7: Diagrama de validación para series de tiempo 33

Ilustración 8: Histograma y Boxplot de potencia generada por Totoral 39

Ilustración 9: Histograma y Boxplot de potencia generada por Canela 39

Ilustración 10: Diagrama de dispersión de Canela de lags de 1 a 9 40

Ilustración 11: ACF y PACF de Canela 41

Ilustración 12: ACF y PACF de Totoral 42

Ilustración 13: Resultados de métodos estadísticos para Canela 43

Ilustración 14: Resultados de métodos estadísticos para El Totoral 44

Ilustración 15: Resultados de predicción recursiva para Canela 46

Ilustración 16: Resultados de prediccíon recursiva para El Totoral 46

Ilustración 17: Gráfico 1 de potencia eólica vs t+h 50

Ilustración 18: Gráfico 2 de potencia eólica vs t+h 50

Ilustración 19: Valores de pesos de atención 51

Ilustración 20: Resultados de predicción para Canela 52

Ilustración 21: Resultados de predicción para El Totoral 52

INDICE DE TABLAS

Tabla 1: Generación bruta por fuente [11] 14

Tabla 2: Funciones de activación 21

Tabla 3: Funciones de pérdida por tipo de problema 23

Tabla 4: Dominio de búsqueda de hiperparámetros 35

Tabla 5: Hiperparámetros constantes 35

Tabla 6: Tabla resumen 38

Tabla 7: Dickey-Fuller Test 41

Tabla 8: Parámetros estructurales de modelos estadísticos 42

Tabla 9: Resultados de modelo propuesto 45

Tabla 10: Resultados del estado del arte 45

Tabla 11: RMSE y MAE para distintos lags 47

Tabla 12: RMSE y MAE para distintos tamaños de kernel 48

Tabla 13: RMSE y MAE para distintas profundidades de convoluciones 48

Tabla 14: RMSE y MAE para distinta cantidad de neuronas por capa 49

Tabla 15: RMSE y MAE para distintas profundidades de capas feed forward 49

Tabla 16: Resultados búsqueda aleatoria de Canela 59

Tabla 17: Resultados búsqueda aleatoria de El Totoral 60

Tabla 18: Resultados de CNN con atención para datos de Canela 61

Tabla 19: Resultados de CNN para datos de Canela 61

Tabla 20: Resultados de red Feed Forward para datos de Canela 61

Tabla 21: Resultados de LSTM para datos de Canela 62

Tabla 22: Resultados de GRU para datos de Canela 62

Tabla 23: Resultados de CNN con atención para datos de El Totoral 62

Tabla 24: Resultados de CNN para datos de El Totoral 63

Tabla 25: Resultados de red Feed Forward para datos de El Totoral 63

Tabla 26: Resultados de LSTM para datos de El Totoral 63

Tabla 27: Resultados de GRU para datos de El Totoral 64

INTRODUCCIÓN

El reemplazo de energías tradicionales por energías renovables es una tendencia en todo el mundo, incluyendo Chile que ha decidido aumentar gradualmente la participación de estas energías en el Servicio Eléctrico Nacional. En particular la energía eólica ha mostrado grandes ventajas por la geografía del país y su bajo costo de operación. Sin embargo, la naturaleza estocástica del viento representa un desafío para la gestión de las plantas eólicas.

En esta memoria se presenta una propuesta basada en redes neuronales convolucionales con mecanismos de atención para pronosticar la potencia eólica generada por parques eólicos chilenos con un horizonte de predicción de 24 horas. Los resultados obtenidos se compararán con modelos del estado del arte, tanto modelos estadísticos y modelos de machine learning.

La memoria presente se organiza de la siguiente manera: En el Capítulo 1 se presenta la problemática y se listan los objetivos de la memoria, en el Capítulo 2 se presentan los distintos modelos que se usarán para realizar la predicción, en el Capítulo 3 se presenta la propuesta de red de atención convolucional, en el Capítulo 4 se muestran los experimentos realizados y en el Capítulo 5 se concluye el trabajo realizado y se deja la puerta abierta a investigación futura.

1. DEFINICIÓN DEL PROBLEMA

La energía eléctrica es hoy en día el alimento que permite que los países se desarrollen. Mantiene a las industrias y a la economía moderna, así como a hospitales y numerosos servicios necesarios para el vivir del día a día[1].

La mayor parte de la energía eléctrica es generada por fuentes no renovables, como gas natural o petróleo. Aunque estas fuentes son numerosas, no son infinitas, por lo que es necesario encontrar fuentes renovables que puedan continuar con la generación de energía eléctrica en un largo plazo.

El *sobregiro* es una métrica introducida en [2] que permite medir la razón entre el consumo de los recursos del planeta versus la capacidad regenerativa de estos recursos. En [3] se muestra que año a año se aumenta el uso de recursos y desde los finales de la década de 70 el sobregiro es mayor a 1, lo que significa que usamos más recursos de lo que el planeta puede sosteniblemente disponer. Uno de los principales responsables es el consumo de los combustibles fósiles y su huella ambiental. Hoy en día el *día de sobregiro* es medido como el día en el que usamos la cantidad total de recursos sostenibles que produce el planeta [4], de manera de comunicar y alertar sobre la sobredemanda ecológica. En el año presente se muestra que el día de sobregiro ocurrió el 22 de Agosto, lo cual es un desplazo de un mes con respecto al año anterior [5].

Un segundo problema que las fuentes no-renovables traen consigo es el gran daño que se produce en el medio ambiente y en la elevación de la temperatura del planeta[6].

La preocupación creciente por el cambio climático llevó a que el año 2015 la Asamblea General de las Naciones Unidas adoptara la Agenda 2030 para el Desarrollo Sostenible, y entre sus objetivos establece explícitamente “garantizar el acceso a servicios asequibles, fiables, sostenibles y energía moderna para todos”[7]. Al año siguiente, se firma el Acuerdo de París sobre el cambio climático[8], en el cual también se hace énfasis en la transición a energías renovables.

Es por esto por lo que durante los últimos años se han desarrollado nuevos métodos de obtener electricidad desde fuentes de energía renovables no convencionales (abreviadas como ERNC), como lo son la energía solar, energía de biomasas o la energía eólica. Estas fuentes son renovables y amigables con el medio ambiente.

Entre las ERNC, la energía eólica ha llamado el interés por su bajo costo de operación y su baja huella ambiental. Sin embargo, las fuentes alternativas presentan un nuevo desafío. Dado que no tenemos conocimiento de, por ejemplo, la radiación solar o la potencia del aire del día de mañana, es complicado poder planificar cuanta energía se producirá. La naturaleza de estas fuentes pone una barrera al integrarlas al plan energético de un país. Es por esto por lo que cada vez se hacen más importantes buenos pronósticos de manera de aumentar la confiabilidad de las fuentes de energías limpias.

En el caso de Chile, el año 2012 se presentó la Estrategia Nacional de Energía[9], en la cual se pone como objetivo aumentar la participación de energías renovables y obligar una inyección de ERNC mínima de un 8% respecto de la energía total generada. Para el año 2025 el 20% de la energía deberá venir de fuentes renovables no convencionales y para el año 2050 se tiene una meta del 70%[10].

Actualmente el panorama se ve prometedor ya que, según el Reporte Anual de las Empresas Generadoras de Energía en Chile del año 2018[11], un 45.9% proviene de energías renovables. Descartando las hidroeléctricas que trabajan un tamaño mayor a 20MW (ya que no se consideran ERNC), 21.2% de la energía provino de ERNC, como se muestra en la Tabla 1, sobrepasando ya el 20% obligado para el año 2025.

Fuente	Porcentaje
Carbón	37%
Hídrica>20MW	20.7%
Hídrica<20MW	**6.3%**
Gas	18%
Solar	**8.2%**
Eólica	**6.2%**
Biomasa	**2.4%**
Petróleo	0.9%
Cogeneración	**0.2%**
Geotermia	0.3%
Total ERNC	**23.5%**

Tabla 1: Generación bruta por fuente [11]

En Chile, el Coordinador Eléctrico Nacional (CEN) es el organismo que se encarga de coordinar la operación del conjunto de instalaciones del Sistema Eléctrico Nacional y programar la generación eléctrica de manera de que se supla la demanda energética a toda hora. Según el artículo 7-13 de la Norma Técnica de Seguridad y Calidad de Servicio, cada parque eólico debe entregar al CEN un pronóstico a una hora, al día siguiente y semanal[12]. Como cabe de esperar, mientras más lejano el pronóstico más impreciso es, sin embargo, estas estimaciones son necesarias para poder suplir de energía al país y al mismo tiempo cumplir con los límites ya mencionados. Si la estimación no fue correcta y la potencia eólica no es suficiente para suplir lo planificado, es necesario encender otras plantas de generación de energía, lo cual implica un costo para el país.

1.1 Objetivos

1.1.1 Objetivo general

Implementar un modelo para pronosticar la potencia eólica mediante una red convolucional con mecanismos de atención usando series de datos nacionales con el fin de mejorar la planificación de la energía eólica.

1.1.2 Objetivos Específicos

- Implementar una red convolucional unidimensional con mecanismos de atención que pronostique la potencia eólica con un horizonte de predicción de 24 horas.
- Comparar los resultados de la red diseñada con métodos de deep learning del estado del arte.
- Comparar los resultados de la red diseñada con el método estadístico ARIMA.

2. MARCO CONCEPTUAL

2.1 Series de tiempo

Una serie de tiempo es una secuencia cronológica de mediciones u observaciones cuantitativas de algún proceso. Sea x_t una medición bajo estudio en un instante t, entonces $x_1, x_2, x_3, \dots, x_N$ es una serie de tiempo.

En cada instante t la característica de interés se describe mediante una variable aleatoria X_t. Usualmente no se puede conocer la distribución a priori de toda la secuencia de datos.

Una característica de interés de las series de tiempo es la estacionariedad, la cual puede ser de tipo débil o fuerte.

Un proceso se llama estacionario débil cuando tanto la media como la covarianza entre X_t y X_s es invariante en el tiempo, es decir:

1. $\mathrm{E}[X_t^2] < \infty, \forall\, t \in \mathbb{Z}$
2. $\mathrm{E}[X_t] = \mu, \forall\, t \in \mathbb{Z}$,
3. $\mathrm{Cov}(\mathrm{X_s}, \mathrm{X_t}) = \mathrm{E}[(\mathrm{X_s} - \mu_\mathrm{s})(\mathrm{X_t} - \mu_\mathrm{t})^\mathrm{T}] = \mathrm{E}[(\mathrm{X_{s+h}} - \mu_\mathrm{s+h})(\mathrm{X_{t+h}} - \mu_\mathrm{t+h})^\mathrm{T}], \forall s, t, h \in \mathbb{Z}$.

Por otro lado, un proceso es estacionario fuerte si su distribución de probabilidad se mantiene invariante al desplazamiento del índice t, es decir:

$$\mathrm{F}(X_0, \dots, X_K) = \mathrm{F}(X_h, \dots, X_{K+h}). \quad (1)$$

2.2 Modelo ARIMA

ARIMA, o modelo autorregresivo integrado de medias móviles, es uno de los modelos más usados para series de tiempo [13], con el cual una serie de tiempo es modelada en términos de sus valores pasados y shocks aleatorios.

Sea X_t una observación en el instante t y ε_t un proceso de ruido blanco con promedio 0, un modelo MA(q) o Moving Average (media móvil en español) considera el proceso

$$\mathrm{X_t} = \sum_{\mathrm{i}=1}^{\mathrm{q}} \beta_i \, \varepsilon_{t-i} + \varepsilon_t, \tag{2}$$

donde $\beta_\mathrm{i} \in \mathbb{R}^\mathrm{q}$ es un coeficiente. De manera similar, un modelo AR(p) o Autoregression (autoregresión en español) satisface

$$X_t = \sum_{i=1}^{p} \alpha_i \, X_{t-i} + \varepsilon_t, \tag{3}$$

donde $\alpha_i \in \mathbb{R}^p$ es un coeficiente. En otras palabras, un proceso MA modela una serie de tiempo como una combinación lineal de ruidos blancos, mientras que un proceso AR lo hace mediante una combinación de observaciones pasadas. Los parámetros q y $p \in \mathbb{Z} \geq 0$ determinan la cantidad de términos del modelo.

Un modelo más sofisticado que los ya vistos es el modelo ARMA(p, q), el cual es una combinación de los modelos AR(p) y MA(q). En este modelo se asume que X_t es generado mediante la fórmula:

$$X_t = \sum_{i=1}^{p} \alpha_i \, X_{t-i} + \sum_{\mathrm{i}=1}^{\mathrm{q}} \beta_i \, \varepsilon_{t-i} + \varepsilon_t, \tag{4}$$

donde α y β son coeficientes y ϵ_t es un proceso de ruido blanco.

Para asegurar la representación de un modelo es necesario que la serie de tiempo sea estacionaria. Un modelo AR(∞) o MA(∞) puede representar cualquier serie de tiempo estacionaria. En comparación, un modelo ARMA(p, q) puede generar cualquier proceso estocástico estacionario con un número finito de parámetros [14].

Sin embargo, no todas las series de tiempo son procesos estacionarios. Por ejemplo, podrían tener tendencias o estacionalidades. Una manera de manejar correlaciones temporales dentro de la serie es el método de diferenciación, el cual consiste en restar a cada elemento de la serie de tiempo elementos anteriores. Por ejemplo, una diferencia de primer orden de X_t estaría dada por $\nabla X_t = X_t - X_{t-1}$, y una diferencia de segunda orden sería $\nabla^2 X_t = \nabla X_t - \nabla X_{t-1}$. Si la secuencia de $\nabla^d X_t$ es estacionaria, entonces satisface un modelo ARMA(p, q) y se dice entonces que X_t satisface un modelo ARIMA(p, d, q)

$$\nabla^d X_t = \sum_{i=1}^{p} \alpha_i \nabla^d X_{t-i} + \sum_{i=1}^{q} \beta_i \varepsilon_{t-i} + \varepsilon_t, \tag{5}$$

el cual es parametrizado por los términos p, q y d. Cabe notar que un proceso ARMA(p, q) es un caso especial de un proceso ARIMA(p, d, q) donde d es igual a 0.

Si definimos L^k como el *operador lag* tal que $L^k x_t = x_{t-k}$, la función $f_p(L) = 1 - \sum_{i=1}^{p} \alpha_i L^i$ y $g_q(B) = 1 + \sum_{i=1}^{q} \beta_i L^i$, entonces podemos reescribir el modelo ARIMA(p, d, q) como

$$f_p(L)(1 - L)^d X_t = g_q(L)\varepsilon_t. \tag{6}$$

Para pronosticar con un proceso ARIMA(p, d, q) es necesario revertir el proceso de diferenciación. Supongamos una serie de tiempo de secuencias X_t satisface un proceso ARIMA(p, d, q), podemos predecir el diferencial de orden d de una observación en el tiempo $t + 1$ como $\nabla^d \tilde{X}_{t+1}$ y luego predecir la observación $\tilde{X}_{t+1}$ como:

$$\tilde{X}_{t+1} = \nabla^d \tilde{X}_{t+1} + \sum_{i=0}^{d-1} \nabla^i X_t. \tag{7}$$

Cuando existen dependencias a largo plazo, es posible que un modelo ARIMA no entregue buenos resultados. Estos casos se pueden deber a una estacionariedad importante, pero que no es resuelta por una integración de grado mayor a 1 ($d > 1$). En estos casos se utilizan modelos fraccionarios denominados ARFIMA, en donde el parámetro d permite valores fraccionarios. En [15] se usa un modelo ARFIMA para modelar la velocidad del viento, obteniendo los mejores resultados con un horizonte de predicción de 24 horas.

Otra manera de combatir las dependencias a largo plazo es un modelo SARIMA$(p, d, q)(P, D, Q)_s$, el aprovecha términos estacionales en la serie para mejorar el modelamiento. Un modelo SARIMA se define como

$$f_P(L^s)f_p(L)(1 - L^s)^D(1 - L)^d X_t = g_Q(L^s)g_q(L)\varepsilon_t. \quad (8)$$

2.3 Modelo GARCH

En el análisis de series de tiempo se define a la heterocedasticidad condicional como la variabilidad no constante a lo largo de la serie que se da con ciertos patrones condicionales. La heterocedasticidad condicional no se tiene en cuenta en los procesos lineales clásicos, ya que una de las premisas de estos es tener una varianza constante, tanto condicional como incondicional, a lo largo de la serie temporal.

Para solucionar este problema, aparecen modelos en los que la evolución de la volatilidad condicional juega un papel importante. Los modelos más populares son los nombrados modelos con heterocedasticidad condicional autoregresiva (ARCH), los cuales tienen en cuenta estas determinadas características. Los modelos ARCH aparecen en los años 80, propuestos por Robert F. Engle, un economista, estadístico y profesor universitario que mediante estos, dio la posibilidad de poder analizar la volatilidad condicional que presentan la mayoría de las series del mercado financiero, en las cuales aparecen períodos turbulentos, con cambios bruscos, seguidos de períodos de calma con apenas fluctuaciones. Gracias a estos, Engle fue galardonado con el Premio Nobel de Economía el año 2003. Sin embargo, se han descubierto desventajas relevantes en los modelos ARCH, que han dado lugar a un gran número de extensiones de estos, como los modelos GARCH los cuales facilitan el modelado y pueden ajustar mejor los datos.

Un proceso ARCH(q) apunta a predecir la varianza condicional de varianza condicional $\sigma_t{}^2$ de una serie de datos.

$$\varepsilon_t = z_t \sigma_t, \tag{9}$$

$$\sigma_t^2 = \alpha_0 + \sum_{i=0}^{q} a_i \varepsilon_{t-i}^2, \tag{10}$$

donde ε_t es un proceso de ruido blanco en el tiempo t, z_t es un proceso de ruido blanco estandarizado con promedio 0 y varianza 1, σ_t es la raíz de la varianza condicional en el tiempo t y $a_j \in \mathbb{R} \geq 0$ son los coeficientes de los parámetros del modelo ARCH. El hiperparámetro $q \in \mathbb{Z} \geq 0$ controla cuantos coeficientes tiene el modelo.

Una importante extensión del modelo ARCH propuesto en [16] reemplaza la representación AR(p) con una formulación ARMA(p, q). Este modelo es denominado ARCH generalizado o GARCH.

La forma general del modelo GARCH(p, q) puede ser escrita de la siguiente forma:

$$\sigma_t^2 = \alpha_0 + \sum_{i=0}^{q} \alpha_i \varepsilon_{t-i}^2 + \sum_{j=0}^{p} \beta_j \sigma_{t-j}^2, \tag{11}$$

donde $a_i, \beta_j \in \mathbb{R} \geq 0$ son los coeficientes de los parámetros del modelo ARCH. Los hiperparámetros $p, q \in \mathbb{Z} \geq 0$ controla cuantos coeficientes de varianzas previas y ruidos blancos previos tiene el modelo, respectivamente.

En [17] se muestra que un modelo GARCH siempre usa menos parámetros que ARCH. Un modelo ARCH de alto orden puede tener una representación GARCH más parsimoniosa que es mucho más fácil de identificar y estimar. Usualmente un modelo GARCH$(1,1)$ con solo tres parámetros en la ecuación de varianza condicional es suficiente para obtener un buen ajuste. De hecho, en [18] se provee evidencia de que es difícil encontrar un modelo de volatilidad que supere a un simple GARCH(1,1).

2.4 Redes Neuronales Feed Forward

Una Red Neuronal Artificial (ANN, por sus siglas en inglés) es un mecanismo que busca encontrar una función $\vec{f}(\vec{x})$ aproximándola con una función $\overrightarrow{f'}(\vec{x}, \mathrm{W})$ aprendiendo los valores de los parámetros W que resulten en la menor pérdida entre los valores reales y los aproximados.

Una Red Neuronal Feed Forward es un tipo de ANN que consiste en:

- Unidades de procesamiento, también llamadas neuronas. Se agrupan en capas.
- Capa de entrada, la cual recibe los datos.
- Capa escondida, puede haber una o más.
- Capa de salida, la cual retorna la salida esperada.
- Conexiones entre las capas. Generalmente cada conexión es definida por un peso w_{ij}^{s} el cual determina el efecto que tiene la neurona j de la capa s sobre la neurona i de la capa $\mathrm{s}+1$.
- Sesgo w_{i0}^{s}, el cual permite a la red modelar funciones que no necesariamente pasen por el origen.
- Función de activación σ, necesaria para modelar no-linealidades. Se pueden ver algunas funciones de activación usualmente usadas en la Tabla 2.
- Función de perdida $\mathcal{L}$, la cual mide la distancia entre el resultado generado y el resultado esperado.

Función lineal	Función ReLu	Función sigmoidal	Función tangente hiperbólica
$\boldsymbol{\sigma(x) = x}$	$\sigma(x) = 0, \;\; x < 0$ $\sigma(x) = x, \;\; x \geq 0$	$\sigma(x) = \frac{1}{1 + e^{-x}}$	$\sigma(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}$

Tabla 2: Funciones de activación

Estos componentes son mostrados en la Ilustración 1, donde se puede ver la capa de entrada (verde), capas escondidas (azul), capa de salida (rojo) cada una con sus neuronas (círculos) y las conexiones entre ellas (flechas). Las neuronas conteniendo un 1 representan valores de sesgo que permiten a la red modelar funciones que no necesariamente pasen por el origen.

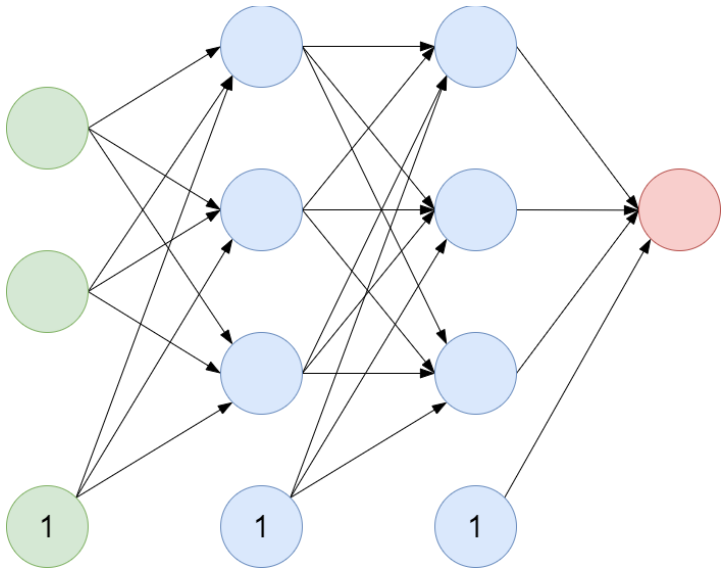

Ilustración 1: Diagrama de red neuronal feed forward

Los valores de input ingresan a la red por las neuronas de entrada y fluye desde la capa de entrada hacia la capa de salida, donde el valor de cada neurona de la red se calcula como

$$a_i^{s+1} = \sigma\left(\Sigma_{j=1}^{N} w_{ij}^{s} x_j^{s} + w_{i0}^{s}\right). \quad (12)$$

A este algoritmo se le conoce como Forward Pass. Al llegar a la capa de salida, es necesario actualizar los pesos w_{ij}^{s} usando el algoritmo Backward Pass. Entonces la regla de actualización se da como

$$w_{ij}^{s} = w_{ij}^{s} - \eta \frac{\partial \mathcal{L}(y,\hat{y})}{\partial w_{ij}^{s}}, \tag{13}$$

donde $\eta \in \mathbb{R}$ es una tasa de aprendizaje que controla en cuanto se actualizan los pesos y $\mathcal{L}$ es la función de pérdida[19]. En la Tabla 3 se especifican las funciones de pérdida más usadas por tipo de problema.

Tipo de problema	Tipo de salida	Función de pérdida $\mathcal{L}$
Regresión	Valor numérico	Error cuadrático medio $\mathcal{L}(y_m, \hat{y}_m) = \sum_m -(y_m - \hat{y}_m)^2$
Clasificación binaria	Dos clases (salida binaria)	Entropía cruzada binaria $\mathcal{L}(y_m, \hat{y}_m) = \sum_m y_m \, ln \, \hat{y}_m + (1 - y_m) \, ln(1 - \hat{y}_m)$
Clasificación multiclase	Múltiples clases (salida en representación one-hot vector)	Entropía cruzada $\mathcal{L}(y_m, \hat{y}_m) = \sum_m \sum_k \left(T(y_m)^{(k)} \, ln(\, T(\hat{y}_m)^{(k)}) + \left(1 - T(y_m)^{(k)}\right) ln\left(1 - T(\hat{y}_m)^{(k)}\right)\right)$ donde $T(y)^{(k)} = I(y = k)$ (función indicatriz)

Tabla 3: Funciones de pérdida por tipo de problema

Para poder calcular el valor de la derivada parcial $\frac{\partial \mathcal{L}(y,\hat{y})}{\partial w_{ij}^{s}}$ es necesario usar la regla de la cadena, donde es posible encontrar problemas de gradiente desvaneciente o gradiente explosivo en redes muy profundas[20].

En [21] los autores proponen un método para predecir la velocidad del viento usando una red de solo dos capas, con la novedad de poder incorporar nuevos datos sin necesidad de recalcular todos los parámetros del modelo. En [22] se presenta una arquitectura en donde se usa un conjunto de redes feed forward para estimar la velocidad del viento en un rango de 6.5 horas, donde la predicción final es una combinación lineal de las salidas de las redes.

2.5 Redes Neuronales Convolucionales

Una Red Neuronal Convolucional (CNN) es un tipo de red neuronal donde los pesos son compartidos a través de los atributos. Estos pesos se agrupan en matrices denominados kernels, y son aplicados mediante una operación de convolución dada por:

$$h_i = \sigma(k * x) = \sigma\left(\sum_{p=0}^{P} x_p \; k_{i-p}\right) \; \forall i \; \in I, \tag{14}$$

donde $k \in \mathbb{R}^{\mathrm{I}}$ es un kernel convolucional, $x \in \mathbb{R}^{p}$ la entrada de la red y $\sigma(\,)$ es una función de activación.

En la Ilustración 2 se puede ver un esquema de convolución unidimensional, donde el kernel de tamaño 3 (borde rojo) opera una operación de convolución a cada secuencia de 3 valores de la entrada.

El efecto de compartir pesos entre características es que cualquier patrón útil que se encuentre en alguna sección de la imagen o conjunto de datos puede ser reusado en otra sección, de esta manera no importa si un patrón está al borde o al centro de una imagen, la red lo podrá encontrar igualmente. A esta propiedad se le llama invariancia a la traslación.

Las CNN son especialmente útiles en problemas donde una característica o variable de entrada no entregan información individualmente. Por ejemplo, un solo píxel de una imagen no entrega mucha información, sino que es su relación con los otros pixeles lo que nos dice que objetos están presentes. Así mismo, un valor en una serie de tiempo (como una palabra en una oración) solo otorga valor cuando se tiene en cuenta otros valores de la serie, por lo que también es un buen uso de las redes convolucionales.

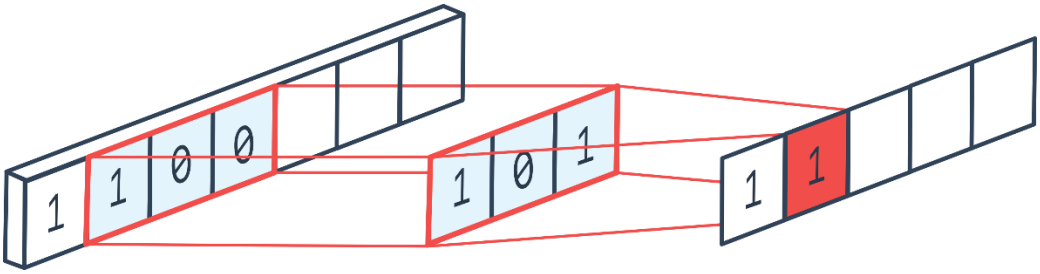

Ilustración 2: Diagrama de convolución 1d

En [23] se menciona que una de las dificultades encontradas al usar redes convolucionales es la necesidad de usar una gran cantidad de capas con tal de poder extraer la información relevante de los datos. Esto acarrea costos computacionales y de hardware, así como los fenómenos de degradación y gradiente explosivo/desvaneciente. Degradación tiene que ver con el aumento de los errores a medida que se aumenta la complejidad de la red. Gradiente explosivo y desvaneciente son problemas que ocurren al actualizar los pesos en el algoritmode backward pass, donde los pesos se comportan de manera normal en las capas más profundas, pero pueden crecer o disminuir exponencialmente a medida en que se actualizan pesos cercanos a la capa de entrada. Para solucionar estos problemas se introduce la arquitectura ResNet o unidad residual, en la cual se propone saltar conexiones entre capas, con tal de ayudar a la red a encontrar la cantidad óptima de capas a entrenar y actualizar los gradientes sin tener problemas por la profundidad de la red. A esta técnica de salto de capas se le llama skip connection. En la Ilustración 3 se puede ver el esquema de una skip connection, en donde el valor de entrada de la red es sumado a la salida de la red. Uno de las desventajas que introduce la arquitectura ResNet es que cada porcentaje de mejora requiere un significativo incremento en la cantidad de capas, lo cual sube linealmente los costos computacionales [24].

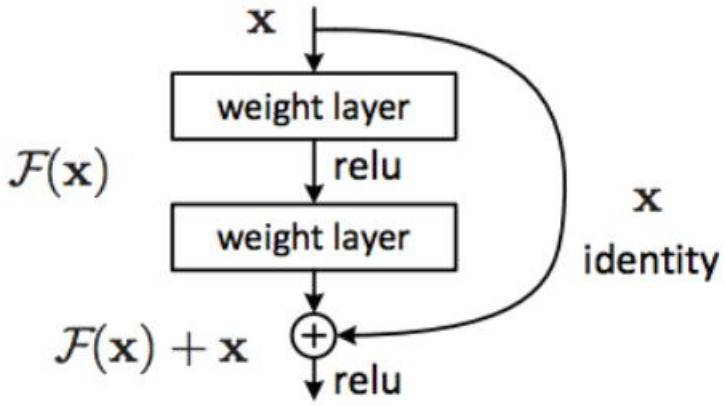

Ilustración 3: Esquema de Skip Connection

En [25] se propone transformar la serie de velocidad del viento con una descomposición wavelet y usar una CNN unidimensional para procesar estos datos y hacer el pronóstico. En [26] usan una red convolucional para procesar las dependencias espaciales y una red feed forward para las dependencias temporales. En [27] se proponen dos arquitecturas de CNN unidimensionales: *CNN 1DSingle*, la cual usa una sola serie de datos unidimensional, *y CNN 1DMultiple*, la cual combina múltiples CNN 1D Single para aprender distintos patrones de información.

2.6 Redes Neuronales Recurrentes

Otro tipo de red son las llamadas Redes Neuronales Recurrentes (RNN), en el cual las conexiones entre sus neuronas presentan ciclos (o retroalimentación). De esta manera, cuando se hace el proceso de forward pass con un valor, este será almacenado en el ciclo y será reutilizado cuando entre otro valor a la red. Estas conexiones permiten describir la salida como una función de las entradas en todos los instantes de tiempo previos. Esto hace que las RNN sean de especial utilidad en problemas de secuencias de tiempo.

Aun cuando estas redes están diseñadas para modelar series de tiempo, acarrean ciertos problemas. Uno de ellos son las dependencias a largo plazo, ya que la red siempre recibe

entradas de datos nuevos, las más antiguas reciben menos atención. Para resolver esto se diseñaron dos arquitecturas basadas en redes recurrentes: redes LSTM (long short term memory) [28] y redes GRU (gated recurrent unit) [29]. Estas redes aprenden mediante operaciones lógicas en cada instante de tiempo que valores eliminar, actualizar y mantener.

En [30] se presenta una RNN para estimar la potencia eólica generada por una turbina con una ventana desde 1 hora a 288 horas.

2.7 Mecanismos de atención

Los mecanismos de atención fueron introducidos por primera vez en [31], asistiendo a redes neuronales recurrentes en el procesamiento de lenguaje natural. Estos permiten a la red a poner atención dinámicamente a secciones de la entrada que son relevantes para la salida, similar a como nosotros nos concentramos en palabras específicas de una oración u objetos en una imagen.

Para esto, el autor usa una configuración encoder-decoder como se puede ver en la Ilustración 4.

Los datos de entrada son transformados por el encoder en un espacio de alta dimensionalidad, y luego el decoder los transforma al espacio de las variables de salida y realizar la regresión o clasificación. Entre ambas se añade un bloque de atención el cual permite que, en cada paso de tiempo, el decoder mire a cada unidad escondida h_i del encoder y calcule a cuál debe prestarles más atención. El bloque de atención consiste en una red neuronal feed forward dentro de la arquitectura de la red principal. Esta red calcula un vector de contexto $\vec{c}$ por cada valor de salida, el cual nos dirá que unidades h_i son relevantes actualmente. Se aplica una función softmax al vector de contexto de manera de transformar los valores del vector en pesos entre 0 y 1. La salida del bloque de atención son pesos que se multiplican con la entrada del decoder. Sea $h_i \in \mathbb{R}^n$ la salida de la red

recurrente encoder, $s_t \in \mathbb{R}^m$ la salida de la red recurrente encoder, podemos calcular la salida $c_t \in \mathbb{R}^n$ como:

$$e_{ti} = \text{ANN}(h_i, s_{t-1}), \tag{15}$$

$$\alpha_{ti} = \text{softmax}(e_{ti}), \tag{16}$$

$$c_t = \sum_{i=1}^{T} \alpha_{ti} h_i. \tag{17}$$

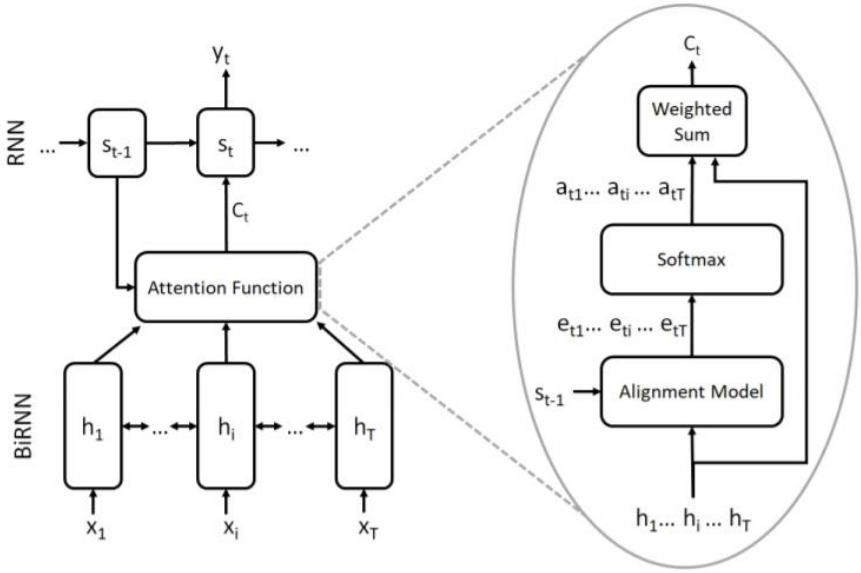

Ilustración 4: Mecanismo de atención en traducción

Aunque el autor inicialmente propuso los mecanismos de atención para traducción de lenguaje usando RNN, es posible llevarlos a otros ámbitos y desprenderlos de las redes recurrentes, como se ve en [32] donde el autor introduce la arquitectura transformer.

La arquitectura CBAM es propuesta en [33] para usar mecanismos de atención en procesamiento de imágenes, en la cual se reemplaza la red neuronal feed forward usualmente usada en mecanismos de atención con una red convolucional. Para imágenes con colores se recomienda usar dos bloques de atención convolucional, uno buscando patrones a través de los canales de la imagen usando convoluciones en una dimensión, y el

segundo buscando patrones a través de los pixeles de la imagen con convoluciones en dos dimensiones.

En [23] se decide mejorar el rendimiento de los bloques de atención en procesamiento de imágenes usando unidades residuales. Cada bloque residual se compone de un número de capas convolucionales conectadas por skip-connections. El autor sugiere usar tantos bloques de atención residuales como canales de entrada tengan las imágenes.

3. PROPUESTA DE SOLUCION

3.1 Caso de estudio

El alcance de este proyecto es aplicar un modelo de red CNN con mecanismos de atención a datos de potencia eólica generadas por las plantas Canela y Totoral. Estas plantas fueron escogidas por ser aquellas con mayor tiempo de funcionamiento, cubriendo desde el año 2008 hasta la actualidad. Esto es importante para un buen funcionamiento de los modelos de redes neuronales.

Canela es la primera planta eólica instalada en Chile en el año 2008, en la cuarta región. Cuenta con 11 turbinas de 1.65 MW, dando una potencia total de 18.15MW. Por otro lado, Totoral, ubicada a 4km de Canela, empezó su funcionamiento el año 2010. Tiene 23 turbinas de 2MW, y una potencia total de 46MW.

3.2 Propuesta

Se busca pronosticar la potencia eólica con un horizonte de predicción de 24 horas, para lo cual se usó un esquema de pronóstico recursivo como se ve en la Ilustración 5. Se usa un conjunto de T datos para pronosticar el dato $T+1$, el cual luego se añade a las variables de entrada con tal de pronosticar el siguiente dato $T+2$ y el resto de los datos a pronosticar. En nuestro caso, se pronosticó hasta el dato $T+24$, lo cual equivale a un día.

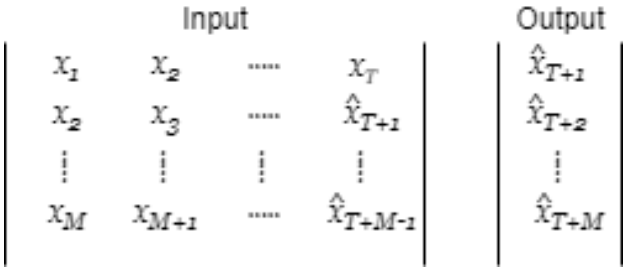

Ilustración 5: Esquema de datos de entrada y salida

Se usó principalmente capas convolucionales unidimensionales para capturar patrones en vecindades de la serie de tiempo. La arquitectura de la red propuesta (Ilustración 6) usa un encoder compuesto de capas convolucionales con el objetivo de encontrar relaciones entre los datos temporales y de llevar los datos a un espacio de alta dimensionalidad, un decoder compuesto de capas feed forward, las cuales serán responsables de la regresión del siguiente dato a pronosticar. Entre el encoder y el decoder se encuentra un bloque de atención basado principalmente en modelos de atención convolucional del estado del arte.

El bloque de atención consiste en un número de unidades residuales convolucionales anidadas. Similar a las ideas tras ResNet, si los saltos entre conexiones permiten modelar un mapeo de identidad, entonces un modelo con unidad residuales tendrá igual o mejor rendimiento que un modelo sin ellas. Siguiendo las sugerencias del estado del arte se usó solo un bloque de atención, ya que nuestra data unidimensional solo cuenta con un canal de entrada.

Sea $h_i \in \mathbb{R}^n$ la salida del encoder convolucional, $K_s \in \mathbb{R}^k$ un kernel convolucional en la capa s, $v_s \in \mathbb{R}^N$ un vector de pesos en la capa s y c_s su respectivo bias, entonces la salida de un bloque de atención convolucional con un elemento residual esta dado por:

$$e_{ti} = \sigma(K_s * h_i + h_i), \tag{18}$$

$$\alpha_{ti} = \mathrm{softmax}({v_s}^T e_{ti} + c_s), \tag{19}$$

$$\mathrm{c_t} = \sum_{i=1}^{\mathrm{T}} \alpha_{\mathrm{ti}}\, \mathrm{h_i}. \tag{20}$$

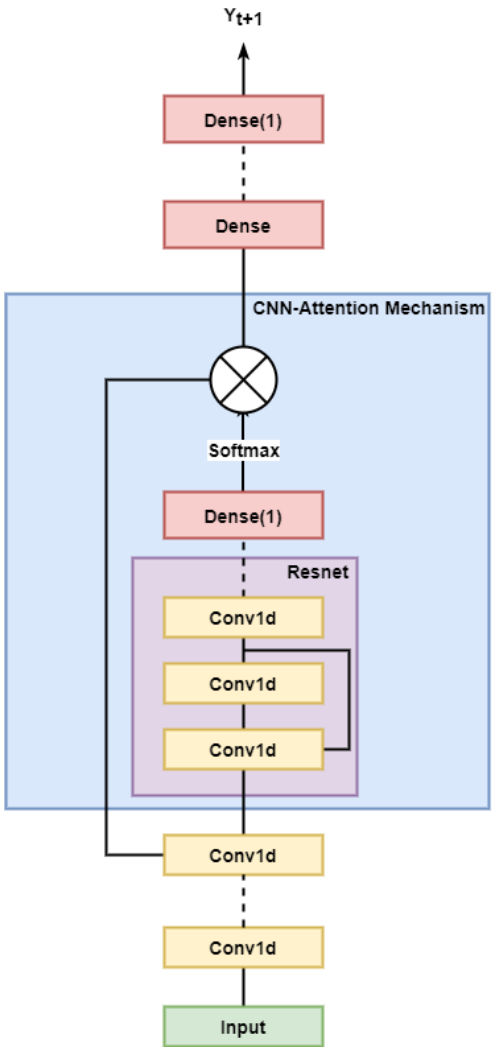

Ilustración 6: Diagrama de propuesta de arquitectura CNN-Attention

La cantidad de convoluciones, unidades residuales y capas feed forward serán hiperparámetros a configurar.

3.3 Validación del modelo

Como se explica en [34], métodos de evaluación de modelos de clasificación y regresión del tipo K-fold cross-validation no pueden ser usados en series de tiempo. Por lo tanto, es necesario un procedimiento alternativo para evaluar el rendimiento de nuestro modelo.

Usaremos la idea abordada en [35] denominada *nested cross-validation*, la cual propone dividir la serie de tiempo en N intervalos de datos y evaluar el modelo con cada intervalo, con la diferencia de que en nuestro caso usaremos una ventana de tamaño constante la cual se moverá por los datos de la serie de tiempo (en lugar de aumentar el tamaño de la ventana, como se propone en el trabajo original) como se ve en la Ilustración 7.

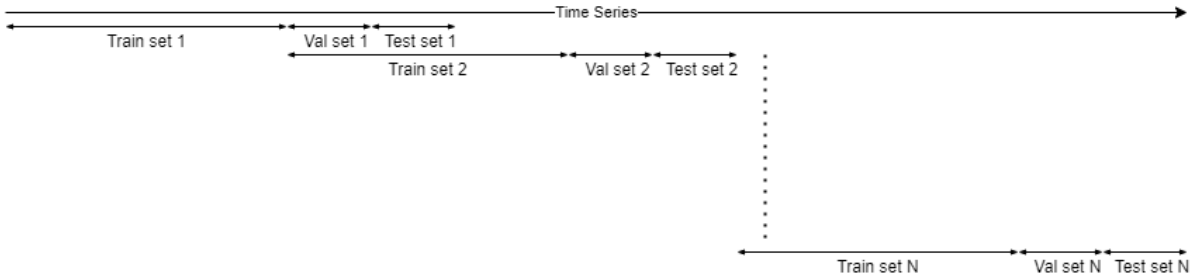

Ilustración 7: Diagrama de validación para series de tiempo

Cada intervalo es luego dividido conjuntos de *entrenamiento* y *pruebas*. A diferencia de [35], añadimos un tercer conjunto de *validación*, con el motivo de ajustar hiperparámetros tales como la cantidad de capas, neuronas por capa, etc. El objetivo de cada conjunto es el siguiente:

- Conjunto de entrenamiento: Ajuste de pesos del modelo.
- Conjunto de validación: Ajuste de hiperparámetros.
- Conjunto de pruebas: Evaluación del desempeño del modelo.

Para cada intervalo de datos se medirán las métricas de Root Mean Squared Error (RMSE) y Mean Absolute Error (MAE) para cada horizonte de predicción desde $t + 1$ hasta $t + 24$.

Adicional a este procedimiento de validación, una técnica de regularización conocida es el early stopping, o detención temprana en español, la cual consiste en detener el entrenamiento del modelo una vez que cierta condición se cumpla. Este procedimiento es especialmente útil en combatir problemas de sobregeneralización[36]. Varias métricas se

pueden usar como parámetros de early stopping. El procedimiento a usar está basado en [37], donde se sugiere usar la pérdida de mínimos cuadrados sobre el conjunto de validación en cada época. Si la perdida registrada en una época es mayor a la registrada en la época anterior se detiene el entrenamiento. Esto nos da aún más motivaciones en usar un conjunto de validación.

3.4 Búsqueda de hiperparámetros

Como se mencionó en la sección 2, las redes neuronales dependen de una gran cantidad de parámetros e hiperparámetros. Mientras que los parámetros son ajustados por el algoritmo de back-propagation, es necesario buscar métodos o heurísticas de búsqueda de hiperparámetros. Los algoritmos más mencionados en la literatura son búsqueda en grilla y búsqueda aleatoria. Ambos algoritmos requieren de un conjunto o dominio en donde se buscarán los hiperparámetros.

El algoritmo de búsqueda en grilla iterará por cada combinación de hiperparámetros buscando el óptimo basado en alguna métrica (usualmente el RMSE). Esto asegura que el óptimo será encontrado, pero es computacionalmente ineficiente. Búsqueda aleatoria elige un número de combinaciones aleatoriamente a probar, por lo que es considerablemente más barato que una búsqueda completa. En [38] se demuestra teórica y empíricamente que la búsqueda aleatoria es más eficiente que la búsqueda en grilla.

Se usó búsqueda aleatoria, los dominios a usar se muestran en la Tabla 4. De manera de alivianar el proceso de búsqueda, los hiperparámetros de numero de kernels, tamaño de pooling, función de activación y paciencia serán mantenidos constantes con los valores mostrados en la Tabla 5. Los valores de hiperparámetros utilizados pueden ser encontrados en el Apéndice A.

Lag	$[1, 200] \in \mathbb{Z}$
Cantidad de convoluciones	$[1, 5] \in \mathbb{Z}$
Tamaño de kernel	$[2, 200] \in \mathbb{Z}$
Cantidad de capas Feed Forward	$[1, 5] \in \mathbb{Z}$
Cantidad de neuronas	$[10, 200] \in \mathbb{Z}$
Learning rate	$[0.00001, 0.001] \in \mathbb{R}$

Tabla 4: Dominio de búsqueda de hiperparámetros

Numero de kernels	32
Tamaño de Pooling	4
Función de activación	Relu
Paciencia	50

Tabla 5: Hiperparámetros constantes

Luego de realizar el proceso de búsqueda seleccionamos el o los modelos que hayan retornado la menor raíz del error cuadrático medio ($RMSE_{t+1}$) en la predicción del dato $t + 1$, la cual está dada por:

$$\mathrm{RMSE}_{t+1} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}\left(y_{n,t+1} - \hat{y}_{n,t+1}\right)^2}. \tag{21}$$

Se hará una segunda búsqueda, esta vez variando un parámetro a la vez, manteniendo el resto con los hiperparámetros seleccionados en la búsqueda aleatoria. En caso de encontrar un RMSE menor, seleccionaremos esa configuración e iteraremos otra vez el proceso de búsqueda. Este segundo proceso de búsqueda se denomina Greedy Algorithm o Algoritmo Voraz.

3.5 Preprocesamiento de los datos

Se usó el esquema presentado en la Ilustración 7, usando $N = 10$ intervalos. Cada intervalo se dividió nuevamente en los conjuntos de entrenamiento, validación y pruebas en una proporción de 60%, 20% y 20% respectivamente. Vale la pena mencionar que los datos de cada intervalo serán normalizados al rango $[0,1]$ usando solo el conjunto de entrenamiento para obtener los parámetros de la normalización.

Luego, la secuencia debe ser dividida en múltiples muestras de entradas y salidas, de manera de que la red pueda aprender la función de probabilidad que mapea las entradas a las salidas. Para esto debemos dividir los intervalos en M muestras, cada una con T pasos de tiempo. Usualmente también es necesario especificar la cantidad de características que contiene cada paso de tiempo, en nuestro caso es solo 1 (la potencia eólica). La red convolucional buscará patrones a través de T valores de potencia consecutivas, usando kernels de convolución de tamaño menor a T.

3.6 Lenguaje de programación

Par la implementación de los modelos tipo ARIMA y GARCH se usó el lenguaje R y en particular la biblioteca *forecast.* La implementación de las redes neuronales se programó en *Python 3* usando los modelos implementados en la biblioteca *Keras* y *Tensorflow* para la

implementación propia de mecanismos de atención. Los experimentos fueron ejecutados en un ambiento Jupyter alojado en el servicio Google Colaboratory.

4. VALIDACIÓN DE LA SOLUCIÓN

4.1 Análisis exploratorio

Como se puede ver en la Tabla 6, la potencia eólica generada por la planta Canela se mueve entre 0 y 68.67, con promedio de 7.1. Para Totoral, se ve una potencia generada menor, la cual se mueve entre 0 y 49.5, con promedio de 4.93.

Es importante destacar que es imposible determinar si los valores de potencia 0 son debido a periodos sin acción del viento o si los generadores estuvieron apagados. Se puede apreciar en la Ilustración 8 e Ilustración 9 que, en ambas plantas, el primer cuartil de datos es muy cercano a 0.

	Canela	Totoral
Count	80158	79486
Mean	12.945	9.111
Std	14.937	10.664
Min	0	0
25%	0.17	0.2
50%	7.1	4.93
75%	21.6	14.6
Max	68.47	49.5

Tabla 6: Tabla resumen

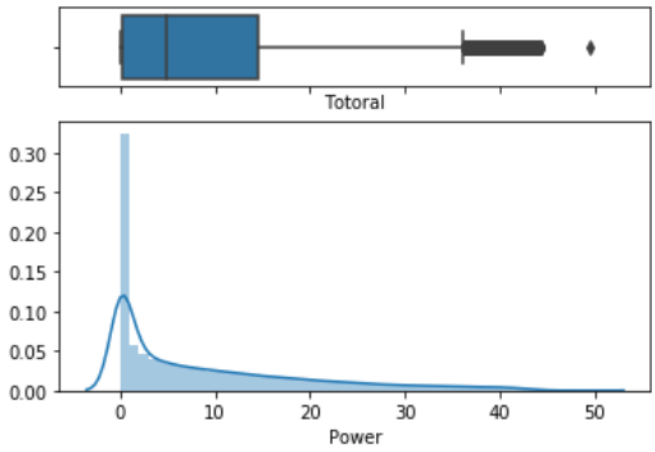

Ilustración 8: Histograma y Boxplot de potencia generada por Totoral

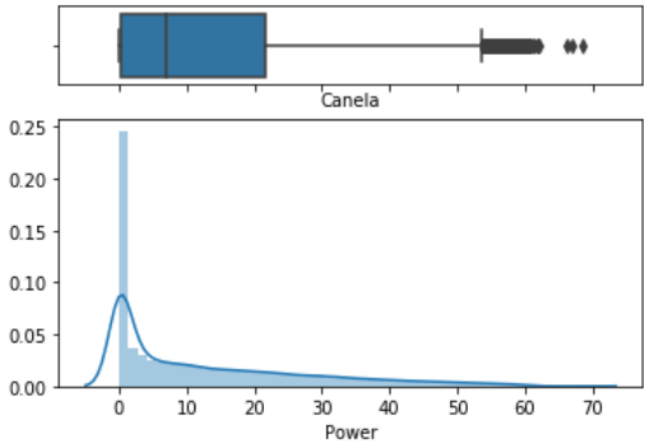

Ilustración 9: Histograma y Boxplot de potencia generada por Canela

Al observar los gráficos de dispersión de varios lags consecutivos (Ilustración 10) podemos ver una dependencia lineal en el primer lag que rápidamente se pierde al aumentar la cantidad de lags.

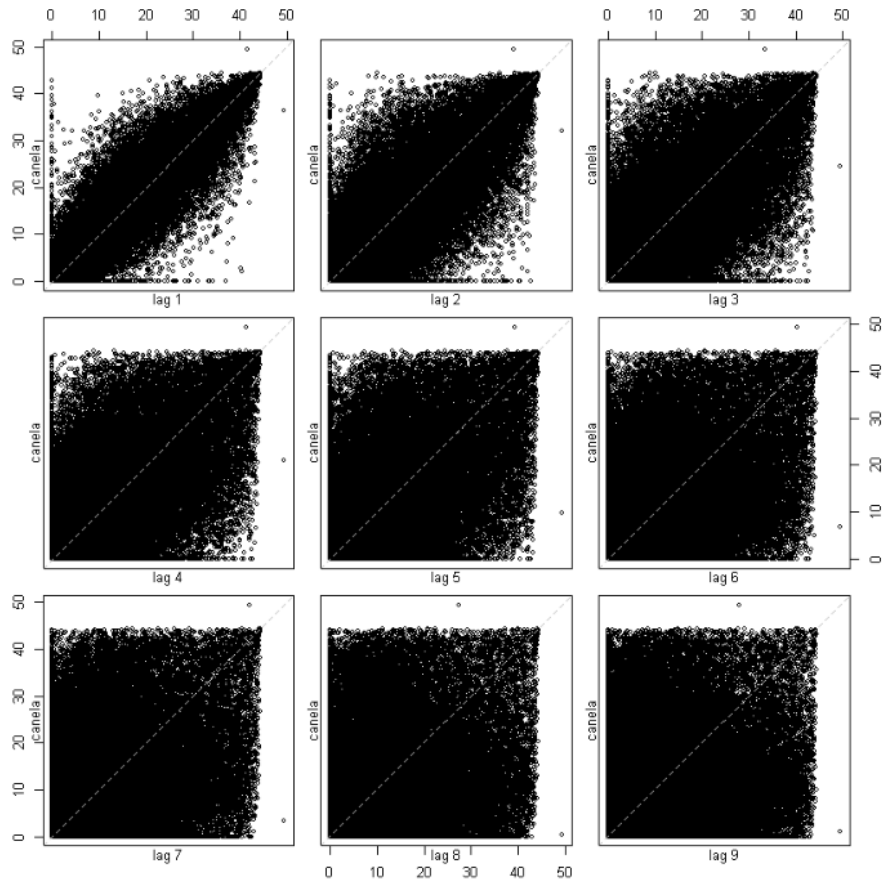

Ilustración 10: Diagrama de dispersión de Canela de lags de 1 a 9

Para saber si es posible modelar una serie de tiempo como un modelo ARIMA se debe primero comprobar que la serie sea estacionaria, para esto podemos hacer un test de Dickey-Fuller. Este test tiene como hipótesis nula que existe una raíz unitaria en una serie de tiempo de una dimensión, y una hipótesis alternativa que depende de lo que buscamos. En nuestro caso, la hipótesis alternativa es que la serie es estacionaria. Para ambas series de tiempos la Tabla 7 muestra que el test falla a la hipótesis alternativa, por lo que las series se pueden trabajar como estacionarias. Dado esto, no es necesario diferenciar la serie para el proceso ARIMA, y por lo tanto podemos usar $i = 0$.

	Canela	Totoral
ADF Estadístico	-27.34	-26.42
Valor p	0.01	0.01
¿Es estacionaria?	Si	Si

Tabla 7: Dickey-Fuller Test

Para entender mejor las características de la serie de tiempo, es importante analizar los gráficos de Autocorrelación y Autocorrelación Parcial (ACF y PACF respectivamente). Principalmente serán estos los que nos ayudarán a encontrar los parámetros del modelo ARIMA. Podemos ver en la Ilustración 11 e Ilustración 12 que ambas series se comportan de manera similar, lo cual tiene sentido al estar ambas centrales eólicas en la misma región. Se puede identificar una frecuencia de 24 horas, lo cual nos sugiere que cada día los patrones de potencia son similares. Mientras que el gráfico de ACF muestra que los primeros 56 lags son significantes para el pronóstico, el gráfico de PACF nos revela que sólo lo es el lag 1. Esto nos sugiera usar un periodo SARIMA con periodo 24 y $p = 1$.

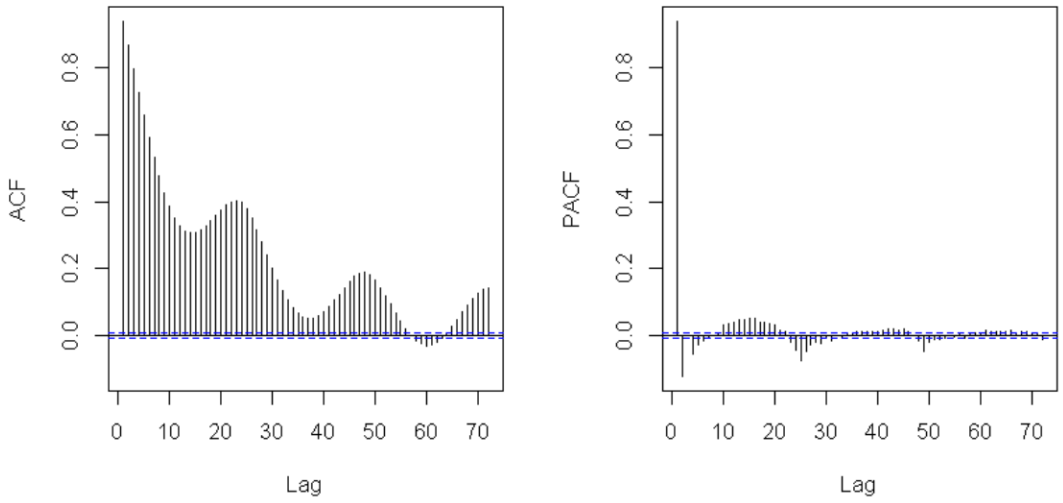

Ilustración 11: ACF y PACF de Canela

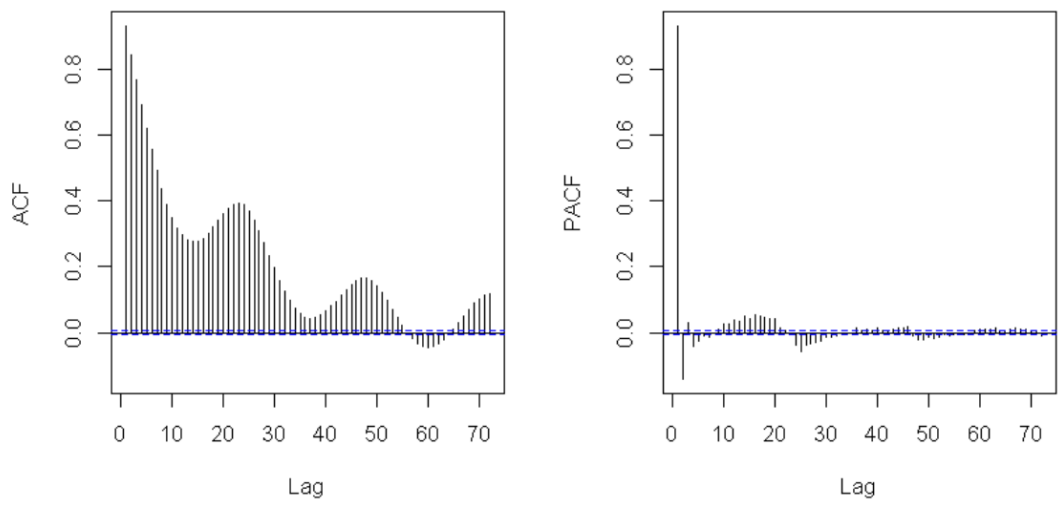

Ilustración 12: ACF y PACF de Totoral

4.2 Resultados métodos estadísticos

Para encontrar los parámetros estructurales se hizo una búsqueda en grilla con los valores mostrados en la Tabla 8, usando la información encontrada en los gráficos de ACF y PACF en la sección 4.1. Tanto para los datos de Canela como de El Totoral se usó un parámetro $d = 0$ y en el caso de SARIMA $S = 24$.

ARIMA		SARIMA		GARCH	
p	[0,4]	**P**	[0,4]	**p**	[0,4]
d	0	**D**	[0,4]	**q**	[0,4]
q	[0,4]	**Q**	[0,4]		
		S	24		

Tabla 8: Parámetros estructurales de modelos estadísticos

Después de realizar la búsqueda, se seleccionaron los siguientes modelos para cada conjunto de datos:

- Canela
  - $\mathrm{ARIMA}(2,0,2)$
  - $\mathrm{SARIMA}(2,0,0)(2,1,0,24)$
  - $\mathrm{GARCH}(2,2)$
- El Totoral
  - $\mathrm{ARIMA}(2,0,2)$
  - $\mathrm{SARIMA}(2,0,2)(1,0,0,24)$
  - $\mathrm{GARCH}(2,2)$

Ya que para ninguno de los modelos ARIMA se utilizó el parámetro d, estos modelos en realidad son ARMA. En el caso del modelo SARIMA de Canela, solo se usó la componente autoregresiva.

En el caso de Canela, se puede ver en la Ilustración 13 que el método GARCH presenta mejores resultados en general, con el modelo ARIMA teniendo mejores resultados de predicción entre los momentos $t+9$ y $t+13$.

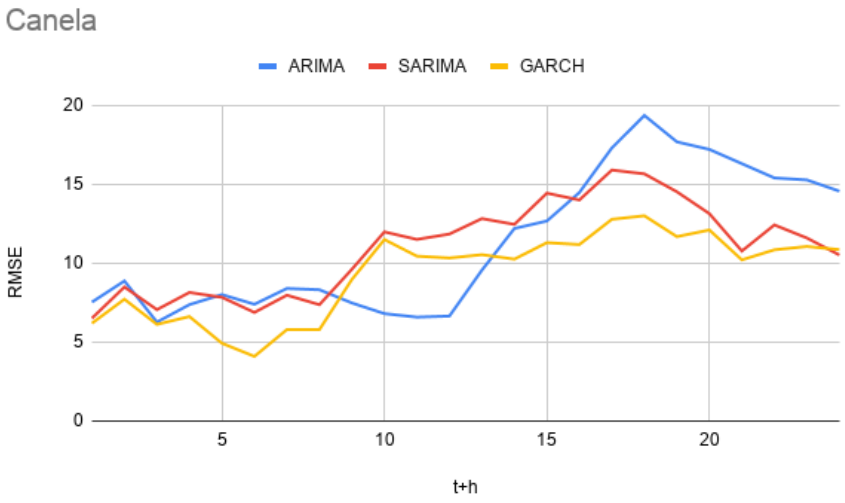

Ilustración 13: Resultados de métodos estadísticos para Canela

En el caso de El Totoral, la Ilustración 14 muestra que se encuentran mejores resultados con GARCH. También se puede ver que los resultados del modelo ARIMA son muy similares a los de SARIMA. Esto puede deberse a que la serie no tiene una componente de estacionalidad tan fuerte como con los datos de Canela.

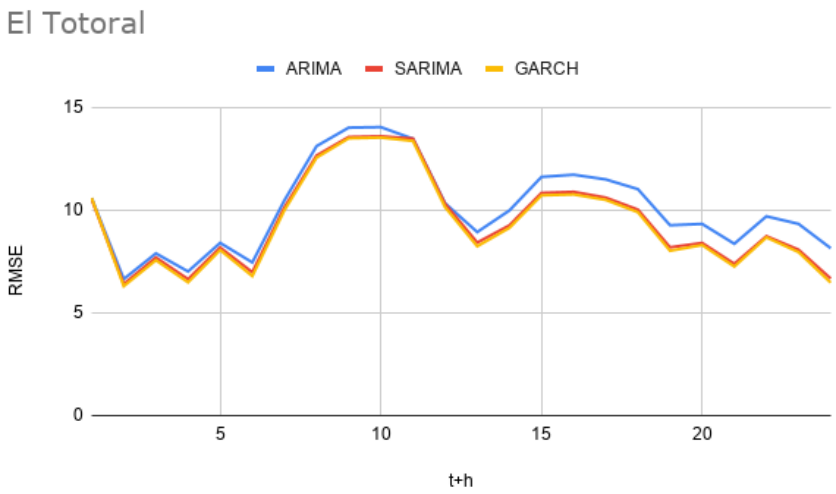

Ilustración 14: Resultados de métodos estadísticos para El Totoral

4.3 Resultados redes neuronales

La Tabla 9 muestra los hiperparámetros de los modelos con menor $RMSE_{t+1}$ obtenidos al aplicar el algoritmo de búsqueda explicado en la sección 3.4 tanto para Canela como para El Totoral. Se puede observar que la profundidad total de la red, el número de neuronas y el coeficiente de aprendizaje óptimos se mantienen igual para ambos conjuntos de datos. La cantidad de lags y el tamaño de los kernels son menores para los datos de potencia eólica de El Totoral.

Estos RMSE obtenidos son menores a los conseguidos por otros métodos del estado del arte, como se puede ver en la Tabla 10.

Data	Lag	Deep Conv	Kernel	Deep FF	Neuronas	Learning rate	RMSE	MAE
Canela	100	1	24	3	100	0,00005	**4,9508**	**3,08442**
El Totoral	48	1	6	3	100	0,00005	**3,8180**	**2,40731**

Tabla 9: Resultados de modelo propuesto

	Canela		El Totoral	
	RMSE	**VAR(RMSE)**	**RMSE**	**VAR(RMSE)**
CNN-Attention	**4,95084**	3.52594	**3,81802**	0.16054
Feed Forward	5.14236	3.53483	3.87378	0.21787
CNN	5.01606	3.46077	3.85158	0.03027
LSTM	4.99188	3.58542	3.87422	0.31916
GRU	4.97749	3.33415	3.90449	0.51267

Tabla 10: Resultados del estado del arte

Los resultados sobre las predicciones al predecir un horizonte completo, desde $t+1$ hasta $t+24$, para Canela en la Ilustración 15 muestran que el modelo de atención propuesto obtiene los mejores errores para predicción en una hora. Esto es consistente con los resultados obtenidos en los experimentos de predicción del dato $t+1$. Entre los tiempos $t+2$ a $t+7$ se desempeña peor que otros modelos, y en ese periodo se obtienen mejores resultados con una red CNN. Entre los tiempos $t+9$ a $t+20$ supera a todos los modelos de machine learning, siendo los errores menores solo para el modelo estadístico Feed Forward, el cual obtiene los mejores resultados para tiempos cercanos al horizonte final de predicción $t+24$.

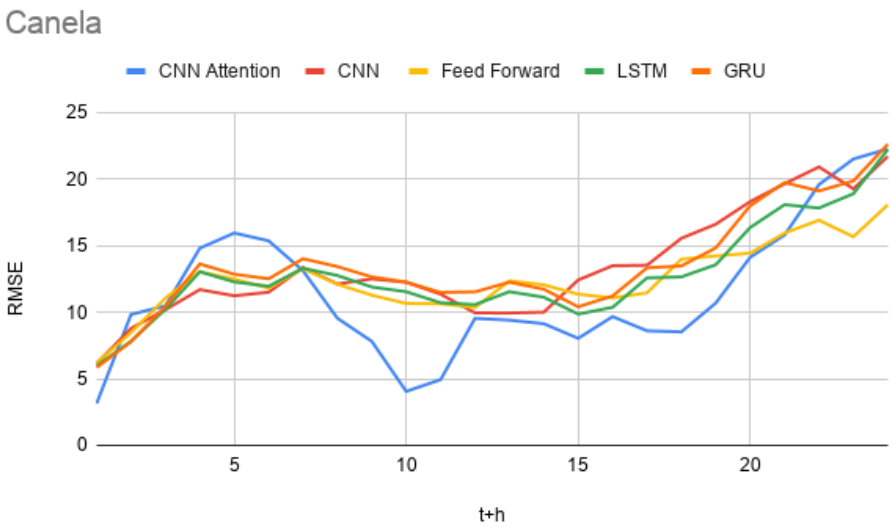

Ilustración 15: Resultados de predicción recursiva para Canela

En el caso de la predicción recursiva para El Totoral, la Ilustración 16 muestra un excelente desempeño de la red de atención convolucional desde $t + 1$ hasta $t + 15$, siendo superada luego por un modelo convolucional y uno feed forward. En el paso de predicción $t + 24$ se ve como el modelo GRU supera a otros modelos.

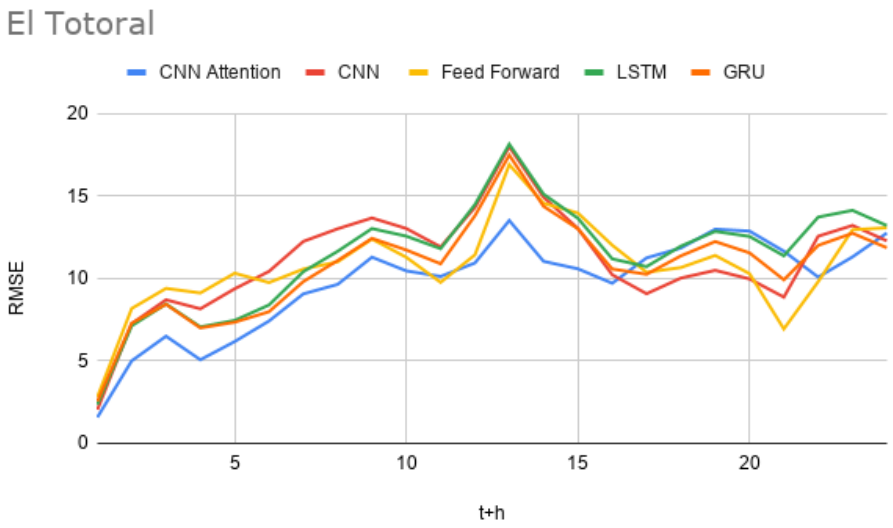

Ilustración 16: Resultados de prediccíon recursiva para El Totoral

Los valores exactos de los resultados pueden ser encontrados en el Apéndice B.

Variar los hiperparámetros de la red de atención convolucional nos puede dar un mejor entendimiento de ella. Al variar la cantidad de lags usados en la Tabla 11, vemos que la red se beneficia de una mayor cantidad de datos para la predicción. En el caso de Canela se obtiene un RMSE menor con una ventana de 100 datos, mientras que para El Totoral el RMSE es menor con 48 datos, lo cual señala una correlación de datos de hasta 2 días, similar al visto en el gráfico de ACF en la sección 4.1.

	Canela		El Totoral	
Lag	RMSE	MAE	RMSE	MAE
1	5,13374	3,20311	3,96094	2,49291
12	5,06988	3,19909	3,93946	2,49982
24	5,04726	3,17825	3,87758	2,50411
48	5,04412	3,11320	**3,81802**	**2,40731**
100	**4,95084**	**3,08442**	3,82472	2,43416
200	5,08780	3,23553	3,89193	2,46000

Tabla 11: RMSE y MAE para distintos lags

Al variar el tamaño del kernel convolucional, en la Tabla 12, se observa que para Canela la cantidad óptima es 24, por lo que la red está encontrando patrones relevantes en vecindarios de 24 pasos de tiempo. Esto, se puede intuir, está relacionado con el periodo diario de la potencia eólica. Para El Totoral el tamaño de kernel con menor RMSE es de 6, menor que para Canela.

	Canela		El Totoral	
Tamaño kernel	RMSE	MAE	RMSE	MAE
6	5,00331	3,10618	**3,81802**	**2,40731**
12	4,98256	3,10593	3,89021	2,44318
24	**4,95084**	**3,08442**	3,82472	2,43416
48	5,05033	3,28487	4,00047	2,57959
100	5,07121	3,29945	3,96156	2,56560

Tabla 12: RMSE y MAE para distintos tamaños de kernel

Como se ve en las tablas 10, 11 y 12, se puede ver que tanto para Canela y El Totoral los hiperparámetros óptimos encontrados son los mismos. Se encontró que solo una capa de convolución es necesaria para encontrar el óptimo.

	Canela		El Totoral	
Profundidad de convoluciones	RMSE	MAE	RMSE	MAE
1	**4,95084**	**3,08442**	**3,81802**	**2,40731**
2	5,04869	3,21513	3,86896	2,51195
3	5,24023	3,36792	3,88266	2,45191
4	5,00286	3,19771	3,96067	2,58734
5	5,04428	3,22106	3,96067	2,44470

Tabla 13: RMSE y MAE para distintas profundidades de convoluciones

	Canela		El Totoral	
Cantidad de neuronas por capa	RMSE	MAE	RMSE	MAE
6	5,44950	3,56141	3,95078	2,51348
12	5,24983	3,35609	3,87546	2,45999
24	5,13757	3,27645	3,84401	2,43739
48	5,08055	3,18717	3,82205	2,42278
100	**4,95084**	**3,08442**	**3,81802**	**2,40731**
200	5,17713	3,26498	3,85056	2,46041

Tabla 14: RMSE y MAE para distinta cantidad de neuronas por capa

	Canela		El Totoral	
Profundidad de capas feed forward	RMSE	MAE	RMSE	MAE
1	5,67521	3,77897	4,02493	2,58827
2	5,13684	3,26521	3,95681	2,48625
3	**4,95084**	**3,08442**	**3,81802**	**2,40731**
4	4,98577	3,11152	3,91531	2,44923
5	4,97514	3,13687	3,93514	2,44687

Tabla 15: RMSE y MAE para distintas profundidades de capas feed forward

Al comparar el modelo propuesto con un modelo convolucional con los mismos hiperparámetros pero sin integrar el bloque de atención convolucional, podemos ver en la Ilustración 17 que el mecanismo de atención generaliza mejor la potencia eólica en las primeras 10 horas. En la Ilustración 18 también vemos que la red con atención generaliza bien y la curva de predicción generada es más suave que sin atención.

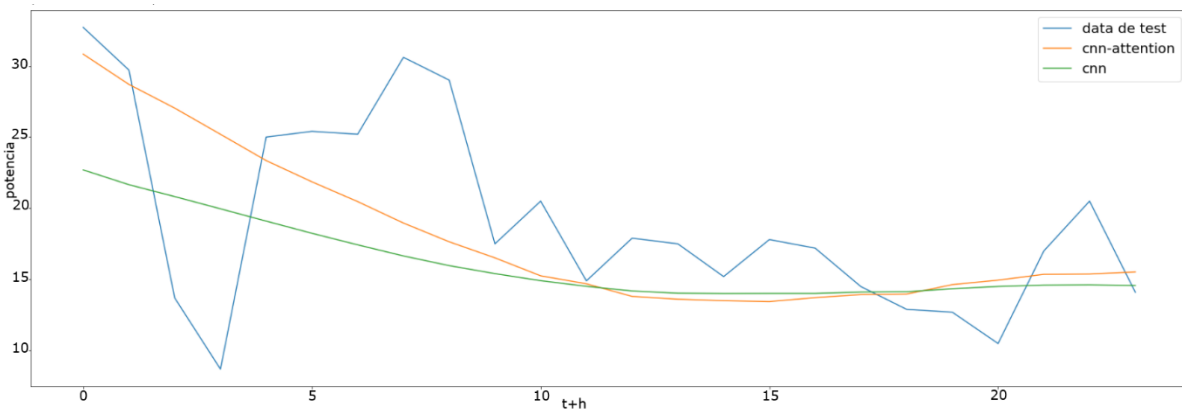

Ilustración 17: Gráfico 1 de potencia eólica vs t+h

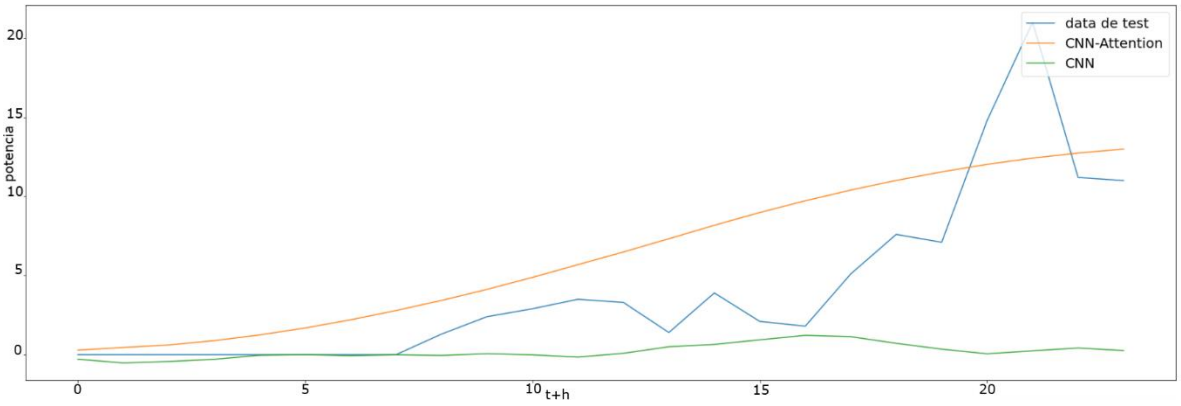

Ilustración 18: Gráfico 2 de potencia eólica vs t+h

Una de las ventajas de los mecanismos de atención es la interpretabilidad de los pesos de atención. En la Ilustración 19 se ve que se usan principalmente los pesos que se multiplicaran con los valores cercanos al lag 24, o sea los más cercanos al valor a predecir.

El resto de los valores son muy cercanos a 0, excepto algunos pesos cercanos al lag 12. Esto es similar a la forma encontrada en los gráficos de PACF en la sección 4.1.

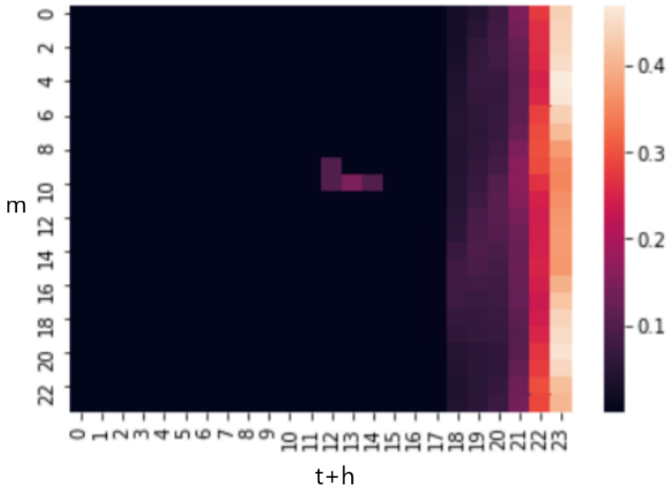

Ilustración 19: Valores de pesos de atención

4.4 Resultados totales

Al juntar los resultados obtenidos para modelos estadísticos y de redes neuronales para los datos de Canela en la Ilustración 20 podemos ver que para el paso $t + 1$ la red de atención convolucional propuesta presenta los mejores resultados, así como también para los pasos $t + 9$ a $t + 19$. Para el resto de los pasos de predicción el modelo GARCH obtiene menores errores.

Para los datos de predicción de El Totoral, vemos en la Ilustración 21 que entre los pasos $t + 1$ a $t + 10$ los mejores resultados se consiguen con el modelo de atención convolucional, mientras que para el resto de pasos es mejor usar el modelo GARCH.

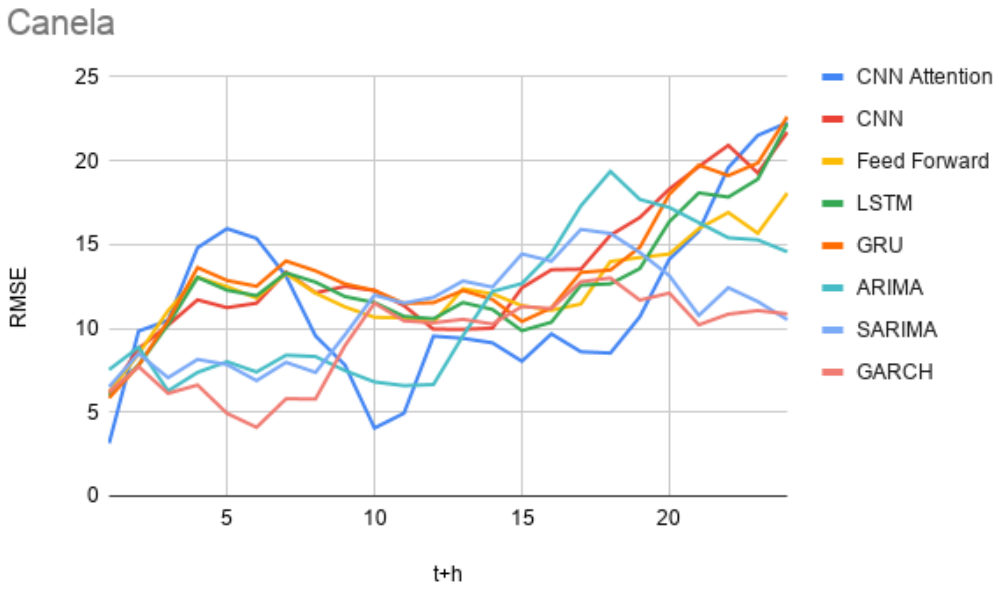

Ilustración 20: Resultados de predicción para Canela

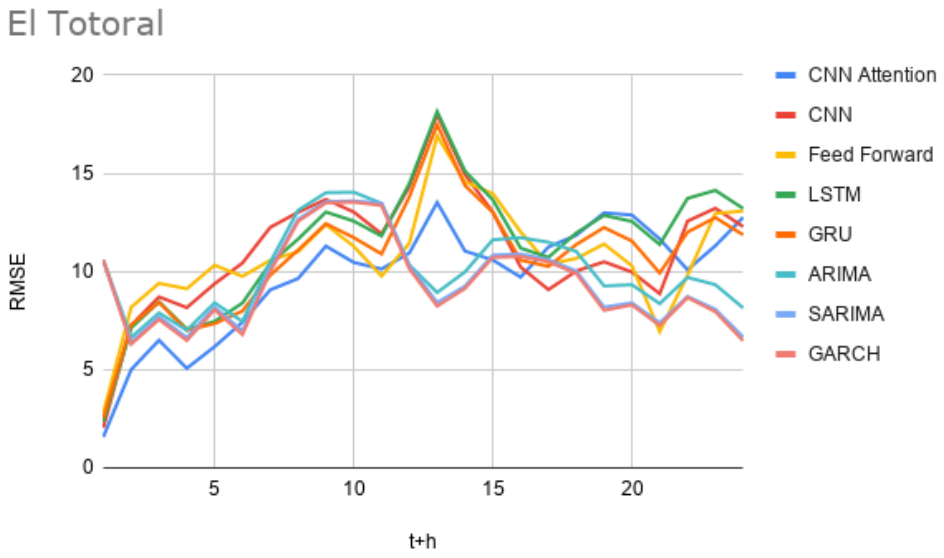

Ilustración 21: Resultados de predicción para El Totoral

5. CONCLUSIONES Y TRABAJO FUTURO

Como se citó en la sección 1.1, el uso de ERNC ha manifestado una tendencia a crecer este último año lo que, con un gran potencial, abre las puertas para realizar investigación relacionada a estas fuentes de energía y facilitar su implementación en la red de energía del Sistema Eléctrico Nacional. La realización de este cambio energético decidirá en gran medida nuestro futuro, pues cada día las consecuencias del cambio climático son más evidentes. Las mejoras sobre la predicción de potencia eólica generada traen además una reducción económica en los precios de la electricidad, puesto que se reducen los costos de gestión y administración de las plantas.

Luego de realizar los experimentos de predicción, podemos concluir que los métodos de redes de atención convolucional muestran el menor RMSE para predicciones entre 1 y 12 horas, mientras que el método estadístico GARCH tienen mejores resultados predicción de 24 horas.

Los modelos basados en redes neuronales tienen la capacidad de realizar un modelo altamente no lineal, lo cual es una fortaleza en este caso por sobre los modelos estadísticos.

De entre los modelos de redes neuronales destaca el modelo de atención convolucional propuesto, puesto que las convoluciones unidimensionales permiten a la red buscar patrones interesantes entre los datos. Las redes convolucionales nacieron como una herramienta para el proceso de imágenes, pero muestran un gran desempeño en otras áreas, como es en este caso la predicción de potencia eólica.

Así mismo, el componente de atención permite a la red saber que datos son importantes para la predicción, y así enfocarse en ellos. Se mostró empíricamente que una red convolucional con un componente de atención muestra mejores resultados que una red sin él.

5.1 Trabajo futuro

Los mecanismos de atención son uno de los grandes avances en redes neuronales del último tiempo y su uso en la predicción de potencia eólica trajo desempeños al nivel de métodos del estado del arte. A futuro sería de interés implementar y probar el uso de mecanismos de atención en otras arquitecturas de red, como redes neuronales GRU o ESN.

En este trabajo se usó una red unidimensional de potencia eólica para la predicción, pero sería de interés observar otras variables que puedan traer ventajas en los desempeños de la red. Para esto sería necesario una investigación a fondo de los datos que afectan la generación de energía de una turbina eólica, como pueden ser la temperatura, época del año, presión ambiental, etc.

La predicción realizada se realizó para un horizonte desde 1 hora hasta 1 día. Sin embargo, el CEN requiere además un pronóstico de una semana. Sería de interés entonces modelar una propuesta que cumpla con este tercer pronóstico requerido basado en el modelo de atención convolucional. Aunque se observó que la red no presenta un buen desempeño a medida que se aumenta el horizonte de predicción en comparación con modelos estadísticos, un modelo de predicción directo en lugar de uno recursivo muestra tener mejores desempeños a través del tiempo, por lo que puede ser una solución al problema a largo plazo.

REFERENCIAS BIBLIOGRÁFICAS

[1] International Energy Agency, "World Energy Outlook Special Report," *World Energy Outlook Spec. Rep.*, 2015, doi: 10.1038/479267b.

[2] H. M. Miller and W. R. Catton, "Overshoot: The Ecological Basis of Revolutionary Change.," *Soc. Forces*, 1982, doi: 10.2307/2578095.

[3] M. Wackernagel *et al.*, "Tracking the ecological overshoot of the human economy," *Proc. Natl. Acad. Sci. U. S. A.*, 2002, doi: 10.1073/pnas.142033699.

[4] G. F. Network, "Earth Overshoot Day," *Glob. Footpr. Netw.*, 2016.

[5] D. Lin, L. Wambersie, M. Wackernagel, and P. Hanscom, "Calculating Earth Overshhot Day 2020: Estimates Point to August 22nd," pp. 1–10, 2020, [Online]. Available: https://www.overshootday.org/content/uploads/2020/06/Earth-Overshoot-Day-2020-Calculation-Research-Report.pdf.

[6] IPCC, *Climate Change 2007 Synthesis Report*. 2007.

[7] "Transforming Our World: The 2030 Agenda for Sustainable Development," in *A New Era in Global Health*, 2018.

[8] C. A. Horowitz, "Paris Agreement," *Int. Leg. Mater.*, 2016, doi: 10.1017/s0020782900004253.

[9] Gobierno de Chile, "Estrategia Nacional de Energía 2012 - 2030 - Energía para el Futuro," *Minist. Energía*, 2012.

[10] M. de E. Chile, "Energía 2050 Política Energética de Chile," *Bulletin of the Atomic Scientists*. 2015, doi: 10.2968/056004010.

[11] Asociacion de Generadoras de Chile, "Reporte Anual 2018."

[12] Comisión Nacional de Energía, "Norma Técnica de Seguridad y Calidad de Servicio para sistemas medianos," p. 183, 2018, doi: http://dx.doi.org/10.1016/j.ejogrb.2012.07.018.

[13] C. Liu, S. C. H. Hoi, P. Zhao, and J. Sun, “Online ARIMA algorithms for time series prediction,” in *30th AAAI Conference on Artificial Intelligence, AAAI 2016*, 2016.

[14] J. D. Hamilton, *Time Series Analysis*. Princeton university press Princeton, 1994.

[15] R. G. Kavasseri and K. Seetharaman, “Day-ahead wind speed forecasting using f-ARIMA models,” *Renew. Energy*, 2009, doi: 10.1016/j.renene.2008.09.006.

[16] T. Bollerslev, “Generalized autoregressive conditional heteroskedasticity,” *J. Econom.*, 1986, doi: 10.1016/0304-4076(86)90063-1.

[17] S. H. Poon and C. W. J. Granger, “Forecasting volatility in financial markets: A review,” *Journal of Economic Literature*. 2003, doi: 10.1257/jel.41.2.478.

[18] P. R. Hansen and A. Lunde, “A forecast comparison of volatility models: Does anything beat a GARCH(1,1)?,” *J. Appl. Econom.*, 2005, doi: 10.1002/jae.800.

[19] C. Robert, “ Machine Learning, a Probabilistic Perspective ,” *CHANCE*, 2014, doi: 10.1080/09332480.2014.914768.

[20] R. Pascanu, T. Mikolov, and Y. Bengio, “On the difficulty of training recurrent neural networks,” in *30th International Conference on Machine Learning, ICML 2013*, 2013.

[21] N. Cheggaga, “Improvements in wind speed forecasting using an online learning,” in *IREC 2014 - 5th International Renewable Energy Congress*, 2014, doi: 10.1109/IREC.2014.6826964.

[22] J. L. Cardenas-Barrera, J. Meng, E. Castillo-Guerra, and L. Chang, “A neural network approach to multi-step-ahead, short-term wind speed forecasting,” in *Proceedings - 2013 12th International Conference on Machine Learning and Applications, ICMLA 2013*, 2013, doi: 10.1109/ICMLA.2013.130.

[23] F. Wang *et al.*, “Residual attention network for image classification,” in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 2017, doi: 10.1109/CVPR.2017.683.

[24] M. Abdi and S. Nahavandi, “Multi-Residual Networks: Improving the Speed and

Accuracy of Residual Networks," pp. 1–8, 2016, [Online]. Available: http://arxiv.org/abs/1609.05672.

[25] H. Liu, X. Mi, and Y. Li, "Smart deep learning based wind speed prediction model using wavelet packet decomposition, convolutional neural network and convolutional long short term memory network," *Energy Convers. Manag.*, 2018, doi: 10.1016/j.enconman.2018.04.021.

[26] Q. Zhu, J. Chen, L. Zhu, X. Duan, and Y. Liu, "Wind speed prediction with spatio-temporal correlation: A deep learning approach," *Energies*, 2018, doi: 10.3390/en11040705.

[27] S. Harbola and V. Coors, "One dimensional convolutional neural network architectures for wind prediction," *Energy Convers. Manag.*, 2019, doi: 10.1016/j.enconman.2019.05.007.

[28] K. Greff, R. K. Srivastava, J. Koutnik, B. R. Steunebrink, and J. Schmidhuber, "LSTM: A Search Space Odyssey," *IEEE Trans. Neural Networks Learn. Syst.*, 2017, doi: 10.1109/TNNLS.2016.2582924.

[29] R. Jozefowicz, W. Zaremba, and I. Sutskever, "An empirical exploration of Recurrent Network architectures," in *32nd International Conference on Machine Learning, ICML 2015*, 2015.

[30] Z. O. Olaofe and K. A. Folly, "Wind power estimation using recurrent neural network technique," in *IEEE Power and Energy Society Conference and Exposition in Africa: Intelligent Grid Integration of Renewable Energy Resources, PowerAfrica 2012*, 2012, doi: 10.1109/PowerAfrica.2012.6498633.

[31] D. Bahdanau, K. H. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 2015.

[32] A. Vaswani *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017.

[33] S. Woo, J. Park, J. Y. Lee, and I. S. Kweon, “CBAM: Convolutional block attention module,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2018, doi: 10.1007/978-3-030-01234-2_1.

[34] C. Bergmeir, R. J. Hyndman, and B. Koo, “Validity of Cross-Validation for Evaluating Time Series Prediction,” *Comput. Stat. Data Anal.*, 2018.

[35] Rob J Hyndman and A. George, “Forecasting: Principles and Practice,” *Princ. Optim. Des.*, 2018, doi: 10.1017/9781316451038.010.

[36] Y. Yao, L. Rosasco, and A. Caponnetto, “On early stopping in gradient descent learning,” *Constr. Approx.*, 2007, doi: 10.1007/s00365-006-0663-2.

[37] L. Prechelt, “Early stopping - But when?,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, 2012, doi: 10.1007/978-3-642-35289-8-5.

[38] J. Bergstra and Y. Bengio, “Random search for hyper-parameter optimization,” *J. Mach. Learn. Res.*, 2012.

APÉNDICE A: RESULTADOS DE BÚSQUEDA ALEATORIA

Lag	Profundidad Convolucion	Tamaño kernel	Profundidad feed forward	Numero neuronas	Learning rate	RMSE	MAE
100	**1**	**12**	**3**	**100**	**0,00005**	**4,95084**	**3,08442**
200	1	20	3	95	0,00005	5,08780	3,23553
95	4	25	3	80	0,00005	5,00286	3,19771
100	1	30	3	20	0,00004	5,13757	3,27645
12	1	24	3	65	0,00005	5,06988	3,19909
120	2	18	1	84	0,00003	5,67521	3,77897
103	2	25	3	12	0,00004	5,24983	3,35609
98	1	30	2	99	0,00005	5,13684	3,26521
24	2	12	1	85	0,00010	5,20694	3,28510
24	3	11	1	10	0,00050	5,10775	3,13340
10	2	15	5	90	0,00020	5,10913	3,19000
12	1	12	5	10	0,00150	5,09311	3,14775
8	5	8	4	20	0,00010	5,00520	3,11871
16	2	8	3	100	0,00055	5,05002	3,15888
200	4	16	3	24	0,00005	5,10688	3,24853
25	4	6	5	25	0,00003	5,01588	3,14513
180	5	50	4	66	0,00053	5,35017	3,47603
150	2	6	4	30	0,00018	5,05710	3,18013
48	2	26	2	48	0,00005	5,30180	3,49285
25	3	10	1	39	0,00004	5,18861	3,26059
90	2	5	2	20	0,00004	5,06808	3,23211
50	2	15	1	24	0,00001	5,29119	3,43864
122	5	4	1	6	0,00002	4,98867	3,14403
93	1	6	3	87	0,00005	5,06407	3,14631
189	1	37	1	74	0,00005	5,63535	3,76963

Tabla 16: Resultados búsqueda aleatoria de Canela

Lag	Profundidad Convolucion	Tamaño kernel	Profundidad feed forward	Numero neuronas	Learning rate	RMSE	MAE
100	**1**	**6**	**3**	**100**	**0,00005**	**3,81802**	**2,40731**
129	1	5	1	80	0,00003	4,02493	2,58827
150	1	8	2	95	0,00004	3,95681	2,48625
72	1	3	4	92	0,00005	3,91531	2,44923
88	1	12	3	70	0,00005	3,90408	2,46806
102	4	9	2	200	0,00010	3,89093	2,46138
200	2	5	1	101	0,00005	4,01350	2,59279
172	2	15	1	90	0,00019	3,96367	2,60757
189	4	24	1	99	0,00004	3,92196	2,52189
24	2	30	4	144	0,00019	3,83668	2,47903
12	5	12	1	150	0,00003	3,88670	2,44527
30	2	18	3	20	0,00002	4,01701	2,54480
50	3	6	4	6	0,00005	3,96789	2,51869
190	4	40	4	77	0,00005	3,97626	2,56125
10	4	8	2	48	0,00005	3,91210	2,43840
200	5	10	5	100	0,00004	3,94572	2,52837
15	4	12	4	12	0,00025	3,85253	2,42541
48	2	66	5	200	0,00010	3,90757	2,46558
24	3	24	5	39	0,00005	3,86993	2,45130
80	1	19	3	139	0,00001	3,82472	2,43416
100	1	100	3	100	0,00019	3,96156	2,56560
70	1	32	3	110	0,00025	4,00047	2,57959
19	1	25	3	139	0,00004	3,87758	2,50411
1	1	1	3	124	0,00015	3,96094	2,49291
95	1	6	3	49	0,00055	3,82205	2,42278

Tabla 17: Resultados búsqueda aleatoria de El Totoral

APÉNDICE B: RESULTADOS DE MODELOS DE MACHINE LEARNING

CNN Att Canela	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8	t+9	t+10	t+11	t+12
RMSE	3.15408	9.83847	10.4801	14.8157	15.9512	15.3703	13.117	9.55163	7.81478	4.0583	4.94228	9.54224
VAR	3.52594	5.30276	5.08586	2.73012	0.40395	0.01995	0.67516	0.21973	1.17726	1.42844	2.40018	4.11963
	t+13	t+14	t+15	t+16	t+17	t+18	t+19	t+20	t+21	t+22	t+23	t+24
RMSE	9.41024	9.14086	8.04299	9.67528	8.60776	8.53022	10.6965	14.1322	15.8015	19.5889	21.5121	22.2694
VAR	5.73526	8.18061	9.68075	9.26218	11.0423	9.13692	8.37826	9.95679	11.9191	10.3953	12.7458	13.2169

Tabla 18: Resultados de CNN con atención para datos de Canela

CNN Canela	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8	t+9	t+10	t+11	t+12
RMSE	6.15185	8.78894	10.1949	11.7051	11.2402	11.5055	13.3565	12.1229	12.5070	12.2596	11.3604	9.95410
VAR	3.53483	4.11724	3.80609	1.23054	1.22095	1.81444	2.78353	3.05708	2.61510	3.41313	3.34761	2.51454
	t+13	t+14	t+15	t+16	t+17	t+18	t+19	t+20	t+21	t+22	t+23	t+24
RMSE	9.94771	10.0092	12.4239	13.4988	13.5333	15.5589	16.6161	18.3134	19.6536	20.9189	19.2708	21.7014
VAR	1.64724	0.05685	0.88566	0.08166	1.67512	0.55599	1.16604	0.51834	2.69374	1.52847	4.43520	5.43322

Tabla 19: Resultados de CNN para datos de Canela

FF Canela	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8	t+9	t+10	t+11	t+12
RMSE	6.15185	8.78894	10.1949	11.7051	11.2402	11.5055	13.3565	12.1229	12.5070	12.2596	11.3604	9.95410
VAR	3.46077	4.06505	3.84998	1.59568	0.75846	0.85818	1.85984	2.05665	1.81280	2.33989	2.65150	1.60020
	t+13	t+14	t+15	t+16	t+17	t+18	t+19	t+20	t+21	t+22	t+23	t+24
RMSE	9.94771	10.0092	12.4239	13.4989	13.5333	15.5589	16.6162	18.3134	19.6536	20.9189	19.2709	21.7015
VAR	0.48670	1.54792	3.33942	3.39815	5.51060	4.31619	4.15693	5.85326	8.36710	7.23486	9.88746	10.5548

Tabla 20: Resultados de red Feed Forward para datos de Canela

LSTM Canela	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8	t+9	t+10	t+11	t+12
RMSE	5.97392	7.82837	10.2301	13.0437	12.2785	11.9554	13.3096	12.7633	11.8991	11.5442	10.7168	10.5767
VAR	3.58542	4.34043	4.37550	2.12667	0.28646	0.86786	1.53954	1.22527	0.43370	0.41072	0.40122	2.00055
	t+13	t+14	t+15	t+16	t+17	t+18	t+19	t+20	t+21	t+22	t+23	t+24
RMSE	11.5445	11.1328	9.8639	10.3718	12.5821	12.6580	13.5730	16.3672	18.0830	17.8338	18.9014	22.2239
VAR	3.47499	5.76057	7.09527	6.94785	9.21222	8.20466	8.42996	11.2671	14.4413	13.6106	16.3438	17.2858

Tabla 21: Resultados de LSTM para datos de Canela

GRU Canela	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8	t+9	t+10	t+11	t+12
RMSE	5.85364	7.79316	10.4475	13.6299	12.8590	12.5152	14.0225	13.4325	12.6544	12.2651	11.4909	11.5355
VAR	3.33415	3.96664	3.96407	1.57291	0.80813	1.41426	2.17776	1.91280	1.17744	1.18183	0.57414	0.92095
	t+13	t+14	t+15	t+16	t+17	t+18	t+19	t+20	t+21	t+22	t+23	t+24
RMSE	12.2628	11.7311	10.4168	11.2233	13.3393	13.4813	14.8372	18.0034	19.7389	19.1151	19.8656	22.6240
VAR	2.31807	4.54159	5.84192	5.67874	7.88608	6.86303	7.12966	9.96115	13.0560	12.0525	14.7455	15.5747

Tabla 22: Resultados de GRU para datos de Canela

CNN Att El Totoral	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8	t+9	t+10	t+11	t+12
RMSE	1.58376	5.00984	6.50389	5.07541	6.17830	7.42528	9.07049	9.64696	11.2979	10.4665	10.1315	10.9458
VAR	0.16054	0.70800	0.16139	0.94145	2.82242	2.25129	0.83611	1.13614	2.54048	3.42697	4.57398	2.75538
	t+13	t+14	t+15	t+16	t+17	t+18	t+19	t+20	t+21	t+22	t+23	t+24
RMSE	13.5235	11.0447	10.5920	9.71675	11.2540	11.8504	12.9934	12.8811	11.6609	10.0895	11.3137	12.7630
VAR	2.31807	4.54159	5.84192	5.67874	7.88608	6.86303	7.12966	9.96115	13.0560	12.0525	14.7455	15.5747

Tabla 23: Resultados de CNN con atención para datos de El Totoral

CNN El Totoral	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8	t+9	t+10	t+11	t+12
RMSE	2.04555	7.26580	8.71224	8.16541	9.38068	10.4346	12.2537	13.0196	13.6757	13.0365	11.9206	14.3142
VAR	0.21787	0.31533	0.74013	2.20775	4.44427	3.85612	1.52632	0.85229	1.52951	1.26504	1.19543	1.56976
	t+13	t+14	t+15	t+16	t+17	t+18	t+19	t+20	t+21	t+22	t+23	t+24
RMSE	17.9966	14.9360	13.0358	10.2306	9.08115	10.0291	10.4991	9.97844	8.87529	12.5684	13.2233	12.2885
VAR	5.07846	3.71402	1.72128	0.18702	1.44916	0.11719	0.19489	0.87590	2.33485	0.73009	0.73057	3.48883

Tabla 24: Resultados de CNN para datos de El Totoral

FF El Totoral	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8	t+9	t+10	t+11	t+12
RMSE	2.81732	8.18527	9.40367	9.12683	10.3253	9.75780	10.5753	11.0150	12.3947	11.2894	9.77105	11.4378
VAR	0.03027	1.11528	0.23189	1.87728	4.80188	4.72728	3.26506	3.17203	3.86217	3.40009	2.90984	0.62903
	t+13	t+14	t+15	t+16	t+17	t+18	t+19	t+20	t+21	t+22	t+23	t+24
RMSE	16.9097	14.5664	13.9749	12.0239	10.3817	10.6680	11.4030	10.2978	6.94423	9.81244	12.9606	13.0816
VAR	4.60646	3.15907	0.92223	0.27903	1.34876	0.15069	0.32672	0.89746	2.74619	0.16152	1.70298	4.74011

Tabla 25: Resultados de red Feed Forward para datos de El Totoral

LSTM El Totoral	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8	t+9	t+10	t+11	t+12
RMSE	2.35723	7.13423	8.45733	7.06761	7.46205	8.39872	10.4183	11.6548	13.0329	12.5721	11.8120	14.4897
VAR	0.31916	0.94343	0.04339	1.53769	3.75247	3.28228	1.67673	1.46341	2.12313	2.18192	2.47513	0.21770
	t+13	t+14	t+15	t+16	t+17	t+18	t+19	t+20	t+21	t+22	t+23	t+24
RMSE	18.1509	15.1169	13.6569	11.1943	10.7250	11.9673	12.8626	12.5560	11.3811	13.7293	14.1376	13.2129
VAR	3.93394	2.93684	0.94581	0.59477	2.15733	0.88791	0.71169	1.61539	3.01992	0.19733	0.98040	3.39249

Tabla 26: Resultados de LSTM para datos de El Totoral

GRU El Totoral	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8	t+9	t+10	t+11	t+12
RMSE	2.55144	7.28665	8.43692	7.00746	7.35263	7.98795	9.81614	11.0937	12.4396	11.7325	10.8969	13.8115
VAR	0.51267	1.07875	0.07697	1.49348	3.70097	3.24816	1.66451	1.38417	2.12033	2.23609	2.56096	0.02238
	t+13	t+14	t+15	t+16	t+17	t+18	t+19	t+20	t+21	t+22	t+23	t+24
RMSE	17.4882	14.3807	13.0222	10.5807	10.2603	11.3780	12.2453	11.5585	9.9274	12.0123	12.7508	11.8700
VAR	3.64835	2.61053	0.71163	0.81099	2.21438	0.92448	0.73659	1.68719	3.15526	0.02903	1.25921	3.74738

Tabla 27: Resultados de GRU para datos de El Totoral