UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE INFORMÁTICA
VALPARAÍSO - CHILE

Aplicación de Redes Recurrentes para la predicción de potencia en plantas eólicas Chilenas

Camilo Valenzuela Carrasco

MEMORIA PARA OPTAR AL TÍTULO DE
INGENIERO CIVIL EN INFORMÁTICA

PROFESOR GUÍA	:	Héctor Allende
PROFESOR CORREFERENTE	:	Carlos Valle

RESUMEN EJECUTIVO

El uso de Energías Renovables no Convencionales (ERNC) en Chile va aumentando año a año, pero la incorporación de estas energías a un Sistema Eléctrico Nacional pensado para la generación de energía tradicional no es simple. La energía eólica es una las ERNC que ha tomado mayor relevancia el último tiempo, no obstante que el comportamiento estocástico de la velocidad del viento hace difícil predecir la potencia eólica generada en el futuro, por lo que es necesario tener modelos de predicción eficientes a distintos horizontes de tiempo.

En este trabajo se realizará una comparación entre los modelos estadísticos tradicionales lineales, con los modelos del estado del arte de las Redes Neuronales Recurrentes (RNN). Mostrando que las RNN obtienen mejores resultados a distintos horizontes de predicción gracias a que son modelos altamente no lineales y no realizan supuestos de la serie al momento de modelar.

Palabras Claves: Energías Renovables, Energía Eólica, ARIMA,Machine Learning, Redes Neuronales Recurrentes .

ABSTRACT

The use of Non-Conventional Renewable Energy (NCRE) in Chile is increasing every year, but the incorporation of these energies into a National Electric System designed for traditional energy generation methods is not simple. Wind power is the NCRE that has become more relevant in recent times, although the stochastic behavior of wind speed makes it difficult to predict the wind power generated in the future, so it is necessary to have efficient prediction models at different horizons.

In this paper a comparison between the traditional linear statistical models, with the state of the art models of the Recurrent Neural Networks (RNN) will be made. Showing that the RNN obtain better results at different prediction horizons thanks to the fact that they are highly non-linear models and do not make assumptions of the series at the moment of modeling.

Keywords: Renewable Energy, Wind Energy, ARIMA, Machine Learning, Recurrent Neural Networks .

GLOSARIO

- **SEN:** Sistema Eléctrico Nacional.
- **CEN:** Coordinador Eléctrico Nacional.
- **CDEC:** Centro de Despacho Económico de Carga.
- **ERNC:** Energías Renovables No Convencionales.
- **NWP:** Numerical Weather Prediction.
- **IID:** Independiente e Idénticamente Distribuida.
- **ARIMA:** Autoregressive Integrated Moving Average.
- **SARIMA:** Seasonal Autoregressive Integrated Moving Average.
- **ANN:** Artificial Neural Network.
- **RNN:** Recurrent Neural Network.
- **ESN:** Echo State Network.
- **LSTM:** Long-Short Term Memory.
- **BPTT:** Back-Propagation Through Time .

Índice de Contenidos

1. Contextualización y Definición del Problema **1**
1.1. Objetivos . . . 4
1.1.1. Objetivo Principal . . . 4
1.1.2. Objetivos Específicos . . . 4

2. Modelos de Predicción **5**
2.1. Modelos Físicos . . . 6
2.2. Modelos Estadísticos . . . 8

3. Análisis de Series de Tiempo. **12**
3.1. Proceso Estocástico . . . 12
3.2. Series de Tiempo . . . 14
3.3. Modelo ARIMA y SARIMA . . . 15
3.3.1. Selección de los parámetros estructurales . . . 17
3.3.2. Estimación de parámetros . . . 19
3.3.3. Predicción del modelo . . . 21

4. Redes Neuronales Recurrentes. **22**
4.1. Arquitecturas de RNN . . . 25
4.1.1. Long Short Term Memory . . . 27
4.1.2. Echo State Network . . . 29
4.2. Preprocesamiento . . . 31
4.3. Selección de Modelo . . . 33

5. Caso de Estudio. **35**

6. Experimentación y Resultados. **37**
6.1. Análisis Exploratorio . . . 38
6.2. Modelos ARIMA y SARIMA . . . 41
6.2.1. Resultados ARIMA y SARIMA . . . 42
6.3. Modelos RNN . . . 45
6.3.1. Experimentos y resultados LSTM . . . 45
6.3.2. Experimentos y resultados ESN . . . 48
6.4. Comparación de Modelos . . . 49

7. Conclusiones y trabajo futuro **51**
7.1. Trabajo Futuro 53

Bibliografía **54**

Índice de Tablas

1.1. Capacidad instalada por tipo de tecnología. Referencia: [8] 2

3.1. Modelo subyacente según el comportamiento de los gráficos de ACF y PACF a distintos rezagos. 19

6.1. Tabla resumen de los distintos Datasets utilizados 38
6.2. Media de RMSE de los modelos $ARIMA$ y $SARIMA$ 44
6.3. Media de RMSE de modelo LSTM con predicción múltiple, variando la cantidad de lags $(t-k)$ de entrada y la cantidad de instantes (*timesteps*) que se desarrolla la RNN en el BPTT. 46
6.4. Media de RMSE de modelo LSTM con predicción recursiva, utilizando 1 lag de entrada y variando la cantidad de instantes (*timesteps*) que se desarrolla la RNN en el BPTT. 47
6.5. Hiper-parámetros seleccionados para modelo LSTM de predicción múltiple 47
6.6. Media de RMSE del modelo ESN con predicción múltiple y recursivo, variando la cantidad de lags de entrada. 48
6.7. Hiper-parámetros seleccionados modelo ESN de predicción múltiple 49
6.8. $p-values$ Test de Hipótesis de diferencia estadística entre MSE de modelos SARIMA y modelos RNN seleccionados, se dice que existe una diferencia estadística si $p-value \leq \alpha$, se utilizó $\alpha = 0{,}01$ 50

Índice de Figuras

2.1. Curva de poder arbitraria de una Turbina de 1 MW [1] 7
2.2. Pasos del modelamiento físico a la predicción de potencia eólica, basado en la figura de [20] 8
2.3. Modelamiento estadístico para la predicción de potencia eólica, basado en la figura de [20] 9

3.1. Ruido Blanco Gaussiano $x_t \sim \mathcal{N}(\mu, \sigma^2)$ 14

4.1. Desarrollo de la RNN recursivamente llegando al instante inicial 22
4.2. RNN Genérica, dependiendo de la arquitectura el flujo de información a los estados $s^{(t)}$ puede variar. 23
4.3. Gráfica de funciones de activación 24
4.4. Red Hopfield con tres nodos. 25
4.5. Red de Jordan 25
4.6. Una *memory cell* del modelo LSTM. El subíndice c denota el valor de la unidad, por lo que se puede expresar los valores de la capa de forma vectorial sin el subíndice. 27
4.7. Echo State Network, la arquitectura está dividida en tres capas, una capa de entrada, un reservorio y una capa de salida. 30

5.1. Potencial Real de las plantas eólicas informada por el CDEC 35
5.2. Mapa de localización de plantas seleccionadas 36

6.1. Boxplot de los distintos Datasets utilizados 38
6.2. Media y varianza mediante una ventana deslizante de tamaño $n = 100$ a través del tiempo en los distintos conjuntos de datos 39
6.3. Diagrama de dispersión de Canela comparando $t - i$ vs $t - j$, en la diagonal $(i = j)$ se encuentra el histograma de los datos. 39
6.4. Gráficos de Auto-correlación y Auto-Correlación Parcial, en el eje x se encuentran los lags o instantes a futuro o pasado y en el eje y la correlación 40
6.5. Media y varianza mediante una ventana deslizante de tamaño $n = 100$ a través del tiempo de datos diferenciados 41
6.6. Comparación entre AIC y Error Cuadrático Medio en Canela 42
6.7. AIC y tiempo de Ajuste dependiendo del Grado del modelo tipo ARIMA 43
6.8. RMSE de los modelos $ARIMA$ y $SARIMA$ 44
6.9. RMSE de cada horizonte de tiempo para el modelo LSTM con predicción recursiva. 47

6.10. RMSE según el horizonte de predicción $t + h$ para el modelo ESN con predicción recursiva. 49
6.11. RMSE según el horizonte de predicción para los modelos seleccionados. 50

INTRODUCCIÓN

El uso de energías renovables no convencionales (ERNC) ha aumentado de forma acelerada en todo el mundo. En Chile se ha instaurado como política pública aumentar gradualmente la participación de estos tipos de energías en el Sistema Eléctrico Nacional (SEN). En especial, la energía eólica ha adquirido una mayor relevancia entre las ERNC en el último tiempo debido al bajo costo que tiene su operación y que no causa mayor daño ambiental.

El ingreso de la energía eólica al SEN es difícil, dado que el sistema fue pensado para energías convencionales. Específicamente, al momento de planificar cómo se suplirá la demanda, es necesario tener una estimación confiable de cuanta energía va a generar cada planta. Debido a la naturaleza estocástica de la velocidad del viento, la predicción de la potencia generada no se puede realizar de forma determinista, por tanto se tienen que utilizar modelos estadísticos o físicos para la predicción.

Esta memoria se centra en comparar distintos modelos estadísticos para la predicción de series de tiempo, comenzando por los modelos tradicionales lineales, para luego utilizar modelos del estado del arte tales como las Redes Neuronales Recurrentes (RNN) y analizar su comportamiento a distintos horizontes de predicción.

La organización de la memoria es la siguiente: En el Capítulo 1, contextualización y definición del problema a abordar. En el Capítulo 2 se presentan los distintos modelos utilizados para realizar la predicción de potencia eólica. En el Capítulo 3 se define el marco teórico y los modelos lineales utilizados en series de tiempo. En el Capítulo 4 se describen los modelos de RNN, sus distintas arquitecturas y algunas técnicas utilizadas para seleccionar el modelo. En el Capítulo 5 se detallarán los distintos conjuntos de datos utilizados y porqué se seleccionaron. En el Capítulo 6 se mostrarán los experimentos y resultados obtenidos para los distintos modelos, y la comparación de éstos. Finalmente en el Capítulo 7 se mencionan las conclusiones de esta memoria y posibles trabajos futuros.

1 Contextualización y Definición del Problema

El uso de la energía eléctrica se ha vuelto cada vez más necesaria para sostener las actividades económicas y fomentar el desarrollo de las naciones. A medida que el desarrollo economía del país crece es necesario generar mayor cantidad de energía; los países industrializados requieren una gran cantidad de energía sólo para mantener su funcionamiento. En el caso de los países en vías de desarrollo, éstos necesitan incrementar su generación de energía para poder crecer económicamente [3].

Tradicionalmente se han utilizado fuentes de energía no-renovables como el petroleo, carbón y gas natural, sin embargo, este tipo de fuentes energéticas dañan el medio ambiente y su uso está relacionado con el cambio climático, afectando al planeta [33]. Al ser fuentes no renovables, existe la posibilidad de que un día se acabe el suministro de este tipo de fuentes energéticas. Dado lo anterior, durante los últimos años se han desarrollado nuevas formas de generar electricidad con fuentes renovables no convencionales como las placas fotoeléctricas, embalses, granjas eólicas, etc.

En Chile, la utilización de energías renovables no convencionales (ERNC) se ha vuelto fundamental para aumentar la capacidad de generación eléctrica y mantener los tratados suscritos en torno al cuidado del medio ambiente como son el Protocolo de Kyoto (1997) y el Acuerdo de Paris (2016). Para cumplir dichos tratados, nuestro país estableció como meta aumentar gradualmente la participación de ERNC en la matriz energética, comenzando con un 20 % en año el 2025 (Ley 20.698) y llegando a un 70 % en el año 2050[9].

Según el reporte anual de las empresas Generadoras de Energía en Chile (tabla 1.1), la capacidad de energía instalada en Chile, en gran parte, utiliza carbón y gas natural, mientras que en relación a las energías renovables existe una gran capacidad de ésta en embalses y centrales de pasada hidráulica (donde se desvía el cause de un río para la generación de energía); No obstante, la cantidad de energía obtenida con nuevas tecnologías todavía está en desarrollo, alcanzando sólo un 10 % entre la energía solar y eólica, ya que dependen de factores que el hombre no puede controlar, como lo es velocidad del viento.

Tipo de Energía	**Potencia [MW]**	**Porcentaje**
Carbón	5.164	23,5 %
Gas Natural	4.633	21,1 %
Derivados del Petróleo	2.972	13.5 %
Cogeneración	18	0,1 %
Embalse	3.402	15,5 %
Pasada	3.223	14,7 %
Biomasa	479	2,2 %
Solar	1.049	4,8 %
Eólico	1.030	4,7 %
Total	21.969	100,0 %

Tabla 1.1: Capacidad instalada por tipo de tecnología. Referencia: [8]

Entre las ERNC, la energía eólica ha presentado recientemente un importante desarrollo, debido a sus bajos costos de operación. A pesar de lo anterior, su integración en sistemas eléctricos presenta numerosos desafíos debido a la naturaleza estocástica del viento.Por ejemplo, la programación de la generación a corto plazo de un sistema eléctrico debe hacer predicciones confiables de cuanta energía eólica será producida para obtener la selección óptima de unidades despachadas, donde los errores de la predicción pueden aumentar los costos de generación del sistema y disminuir su confiabilidad; Por lo que para mantener un suministro eléctrico económico y confiable es necesario pronosticar de manera precisa la generación eólica en diferentes horizontes de predicción.

El organismo encargado de realizar esta programación es el Coordinador Eléctrico Nacional (CEN) en específico el Centro de Despacho Económico de Carga (CDEC), que está encargado de determinar y coordinar la operación en conjunto de instalaciones del sistema eléctrico, de modo que el costo del abastecimiento eléctrico del sistema sea el mínimo posible.

Según el articulo 7-13 de la Norma Técnica de Seguridad y Calidad de Servicio entregada por el CDEC [6], el coordinador de un parque eólico tiene que elaborar y poner a disposición del CDEC, la siguiente información sobre la predicción de producción de energía de la planta:

1. Pronóstico a Corto Plazo: generación horaria prevista para las siguientes doce (12) horas, con actualización horaria.

2. Pronóstico del Día Siguiente: generación horaria prevista para las siguientes cuarenta y ocho (48) horas, actualizada cada seis (6) horas.

3. Pronóstico Semanal: generación horaria prevista para la próxima semana (168 horas siguientes), actualizada cada veinticuatro (24) horas.

Estos pronósticos son fundamentales para que la matriz eléctrica pueda funcionar correctamente, puesto que que los costos de la generación de energía se incrementarán si el error del pronóstico es muy elevado ello se debe a que se necesitan ocupar recursos de otras plantas que no estaban contemplados inicialmente para suplir la falta de energía generada por la planta.

Al momento de realizar el pronóstico de la energía eólica generada se debe tener en cuenta si el modelo predice directamente la potencia, o lo predice indirectamente pronosticando la velocidad del viento en cierto instante de tiempo.

Si se utiliza el pronóstico indirecto hay que tener en cuenta que existen varios factores que afectan el cálculo de la energía producida, dado que la relación entre la rapidez del viento y la potencia generada no es lineal, si no que tiene una dependencia cúbica de la rapidez del viento [16] como se puede ver en la ecuación 1.1, donde ρ es la densidad del aire, A es el área de la turbina y v es la velocidad del viento.

$$P = \frac{E}{t} = \frac{1}{2} A \rho |v|^3. \tag{1.1}$$

Debido a esta relación cúbica entre la potencia generada y la magnitud de la velocidad del viento, un pequeño error en la predicción de la rapidez lleva a un gran error en la energía pronosticada. Es por este último punto que se hace necesario buscar modelos para poder predecir la velocidad del viento de forma eficaz.

En el caso del pronóstico directo, la potencia generada podría depender además de otros factores, como mantención de los generadores, fallas en los equipos o decisiones de la generadora.

1.1. Objetivos

1.1.1. Objetivo Principal

Realizar un estudio comparativo de los modelos de pronóstico lineales generales y no lineales para la predicción de la potencia eólica generada a corto plazo según el artículo 7-13 punto a. de la Norma técnica de Seguridad y Calidad de Servicio del CDEC.

1.1.2. Objetivos Específicos

- Analizar de las series de tiempo de generación de potencia eólica de plantas Chilenas.
- Seleccionar un modelo estadístico del tipo ARIMA y modelos del estado del arte de la familia de las Redes Neuronales Artificiales Recurrentes.
- Comparar los modelos y seleccionar el modelo más eficaz tomando como medida de desempeño el *Root Mean Square Error* (RMSE).

2 | Modelos de Predicción

La predicción de la velocidad del viento ha sido estudiado desde hace muchas décadas. En un comienzo se utilizó para aplicaciones que no estaban relacionadas con la generación energética. En general se utilizaba para la navegación, como la marítima, pero luego se utilizó para pilotear aviones y guiar misiles.

Últimamente, para facilitar la incorporación de las ERNC, específicamente la generación de potencia eólica, ha recobrado importancia la necesidad de disponer de buenas predicciones por lo que se ha recurrido a la experiencia de las aplicaciones antes mencionadas y desarrollar nuevos modelos que permitan mejorar las predicciones realizadas.

La dificultad en la predicción de potencia eólica está en la dependencia temporal que tienen los datos, ya que el tiempo siguiente depende del tiempo anterior. Esta dependencia temporal hace que muchos métodos de regresión o predicción no se puedan usar puesto que asumen como supuesto que la variable a analizar sea **Independiente e Idénticamente Distribuida (IID)**, por lo cual se necesitan modelos de series de tiempo que tomen en cuenta la dependencia temporal de los datos.

Dependiendo de la información que se pretende entregar, según la literatura, la predicción se puede dividir en cuatro horizontes diferentes: Muy corto (Unos pocos segundos a 30 min), Corto (30 min a 6 horas), Mediano (6 a 24 horas) y Largo plazo (24 a 168 horas) [20]. Dependiendo del horizonte de predicción la información puede ser utilizada para propósitos diferentes, en el caso del muy corto plazo esta información es útil para el monitorio de las turbinas y de la potencia generada en cada instante por dichas unidades. La información a corto plazo y mediano plazo sirve para realizar planificación en la matriz energética, y la de largo plazo puede ser utilizada para planificar el mantenimiento de las turbinas.

Los modelos utilizados para realizar la predicción se pueden dividir entre **modelos estadísticos** y **modelos físicos**. Los primeros utilizan datos históricos para realizar la predicción, mientras

que, los modelos físicos utilizan ecuaciones diferenciales parciales para realizar la predicción. Adicionalmente existen modelos híbridos que combinan ambos tipos de modelos para poder obtener mejores resultados.

2.1. Modelos Físicos

Los modelos físicos buscan realizar una predicción indirecta de la potencia eólica mediante la relación que tiene con la velocidad del viento, realizando una modelación fenomenológica que tome en cuenta todos los factores y leyes físicas, por ejemplo, la conservación de la energía y la masa que afectan a la velocidad del viento.

Luego de realizar el modelamiento matemático del fenómeno se realiza una implementación numérica del mismo, también llamada Numerical Weather Prediction (NWP), en donde se divide el dominio global en una grilla y se calcula la dinámica del fenómeno utilizando algoritmos numéricos, que buscan la solución exacta o aproximada del modelo.

Existen diversas implementaciones de modelos de tipo NWP, siendo uno de lo más utilizados un sistema que cuenta con varios modelos para diversas aplicaciones llamado Weather Research and Forecast Model (WRF) [31]. Este software puede realizar la modelación numérica de variables meteorológicas a distintas escalas, y cuenta con un módulo de asimilación para ingresar datos realeas al modelo. Cabe destacar que se tiene que tener en cuenta la altura en la que están las turbinas, ya que la velocidad del viento varía en gran medida según la altura.

Luego de realizar el modelo global del fenómeno, se utilizan métodos de Downscaling para pasar de un modelo global a uno local, con el objetivo de realizar un refinamiento del modelo tomando en cuenta varios factores que no se consideran en el modelo global, como por ejemplo, las especificaciones de las turbinas. Para realizar el Downscaling, al igual que en el caso de los modelos de NWP, existen implementaciones que realizan dicho trabajo como es Wind Atlas Analysis and Application Program (WAsP) [26], siendo necesario contar con información detallada del terreno donde se ubica la planta eólica.

Además del proceso de Downscaling, se necesita tener en cuenta aspectos técnicos de la turbina en sí, puesto que la relación teórica entre la velocidad del viento y la potencia generada está acotada en la práctica, si la velocidad del viento es muy grande esta puede dañar la turbina, por lo cual no es

recomendable utilizarla a grandes velocidades. De lo anterior nace el concepto de *Power Curve*[28], una función utilizada para realizar el cálculo de la potencia generada por una turbina dependiendo de la velocidad del viento . Esta función se divide en tres tramos como se muestra en la Figura 2.1. El primer tramo la velocidad del viento es tan pequeña que no tiene el torque necesario para el funcionamiento de la turbina hasta que llega al punto de *cut-in*, luego la generación de la energía se comporta de forma cúbica hasta que llega a la potencia máxima que puede generar obtenida en la *Rated Speed*, y se mantiene relativamente constante hasta que se llega a un punto denominado de *cut-off* donde la velocidad del viento puede dañar el funcionamiento de la turbina por lo que se apaga automáticamente.

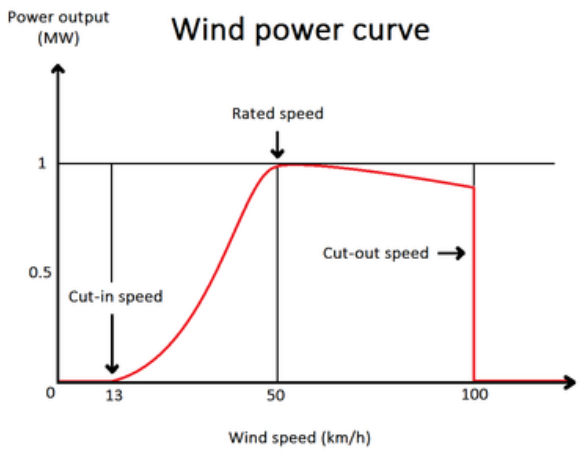

Figura 2.1: Curva de poder arbitraria de una Turbina de 1 MW [1]

Teniendo en cuenta todos estos aspectos, el modelamiento físico de la predicción de potencia eólica se puede dividir en tres pasos: primero realizar la predicción global utilizando un modelo NWP, luego utilizar la información de la planta eólica para realizar el downscaling de la predicción global a una local de la turbina, y finalmente con la velocidad a nivel de la turbina se realiza la predicción teniendo en cuenta la curva de poder de dicha turbina. Este proceso se puede observar en la Figura 2.2.

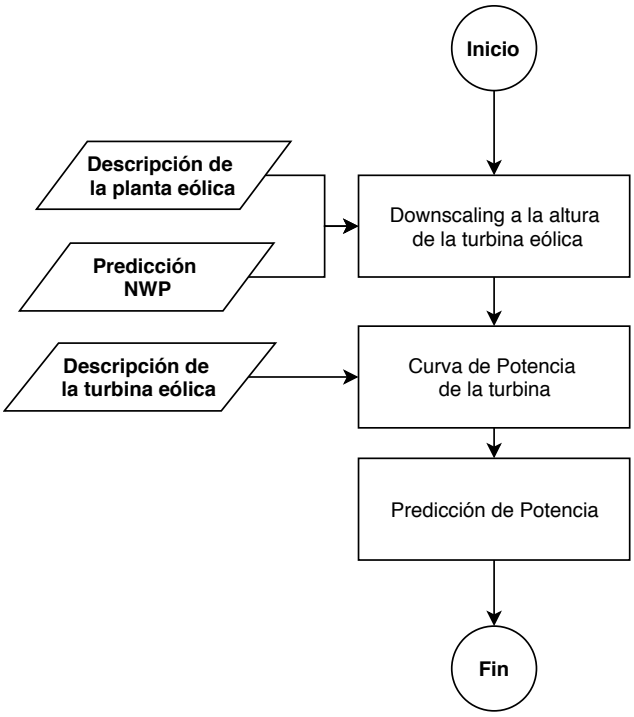

Figura 2.2: Pasos del modelamiento físico a la predicción de potencia eólica, basado en la figura de [20]

Según [32], los modelos físicos como los de la familia NWP son útiles para horizontes de predicción a largo plazo, en cambio para el corto y muy corto plazo los modelos estadísticos tienen mejores resultados. El mayor problema de los modelos físicos está asociado a la necesidad de tener información del terreno para realizar un buen modelamiento, un proceso que puede ser costoso.

2.2. Modelos Estadísticos

Los modelos estadísticos se basan en que la velocidad del viento, y por ende la potencia eólica generada se pueden considerar una variable aleatoria que puede ser modelada como un proceso estocástico. Estos modelos pueden realizar la predicción de la velocidad del viento e indirectamente realizar una predicción de la potencia generada mediante la curva de poder, pero a diferencia de los modelos físicos, pueden realizar la predicción directa de la potencia eólica generada sin necesidad de contar con información del terreno donde se encuentran las turbinas, si no que utilizan datos históricos para realizar la predicción. Una descripción general se puede observar en la Figura 2.3, donde se pueden utilizar datos observados, generados mediante modelos NWP, o ambos para obtener mejores resultados en la predicción.

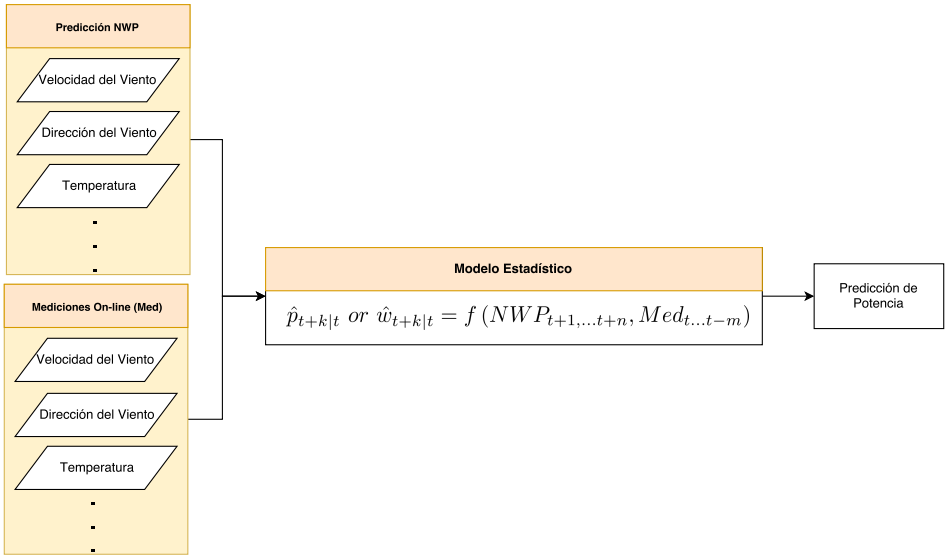

Figura 2.3: Modelamiento estadístico para la predicción de potencia eólica, basado en la figura de [20]

Dentro de los modelos estadísticos tradicionales se encuentran los modelos Autoregresivos Integrados de Medias Móviles (ARIMA), detallados en la sección 3.3, que modelan el proceso mediante un modelo lineal general utilizando datos históricos.

Al momento de modelar una serie mediante un modelo ARIMA, se recurre a la metodología de Box-Jenkins [4]. Esta metodología define cómo encontrar el mejor modelo según algunos criterios teóricos. En la práctica esta metodología puede sufrir ciertas modificaciones: en [11] proponen generar un modelo diferente cada vez que se quiera realizar una predicción utilizando una pequeña cantidad de datos y así reducir el costo computacional de ajustar un sólo modelo con todos los datos históricos. En [15], al momento de seleccionar y validar el mejor modelo ARIMA utilizando la metodología de Box-Jenkins, muestran que existen métricas que reflejan mejor el ajuste cuando se trabaja con series de velocidad del viento y potencia generada.

Además de seleccionar el modelo ARIMA que mejor ajusta a los datos, hay que tener en cuenta si se realizará una predicción directa o indirecta de la potencia eólica. Según [30] la predicción directa da mejores resultados en comparación a la predicción indirecta, debido a que la curva de poder utilizada para realizar la conversión, es una curva ideal, que en la práctica puede ser diferente.

Una característica del modelo ARIMA es realizar predicción a distintos horizontes. En [29] realizan predicción desde 6 horas hasta 24 horas y observan que el modelo tiene resultados satisfactorios hasta las 12 horas, pero no cuando crece el horizonte de predicción a 24 horas. En [25] en

cambio, proponen utilizar un ensamblado de modelos ARIMA para distintos horizontes de tiempo y así mejorar los resultados a un horizonte mayor.

Como los modelos ARIMA no dan buenos resultados cuando existen dependencias a largo plazo, se utilizan modelos denominados *Fractional ARIMA* (ARFIMA), estos modelos buscan mantener las dependencias a largo plazo de la serie. En [21] muestran de forma empírica que estos modelos obtienen mejores resultados con un horizonte de predicción de 24 horas.

Otro factor relevante son las propiedades de la serie. En el modelo ARIMA, la serie a modelar se supone estacionaria. Para los casos cuando la serie no cuenta con esta propiedad es necesario utilizar una variación de este modelo. En [7] proponen el *Limited-ARIMA* (LARIMA), dado que las series de velocidad del viento y potencia están acotadas a un rango, el modelo LARIMA toma en cuenta este factor y limita los valores de salida de la serie, estabilizando la media y varianza del proceso.

En los últimos años además de los modelos ARIMA se comenzaron a utilizar modelos no lineales conectivistas denominados Redes Neuronales Artificiales (ANN). Los modelos ANN generalmente se representan como una grafo con tres capas: una capa de entrada, una capa oculta y una capa de salida, donde los nodos se denominan neuronas. Cada nodo de la capa oculta y de salida realiza una combinación lineal de sus conexiones con las capas anteriores, en se utiliza una función usualmente no lineal para otorgar la capacidad de encontrar dependencias no lineales entre los datos.

A diferencia de los modelos ARIMA, los modelos de ANN no realizan supuestos en la serie de tiempo que modelan. En [5] realizan una comparación entre los modelos ARIMA y de ANN, dada la no linealidad de las ANN, obtienen mejores resultados en algunos conjuntos de datos que no cumplen los supuestos que se realizan en los modelos ARIMA, pero no permiten la construcción de intervalos de confianza.

Otra familia de ANN que se utiliza en problemas de series de tiempo son denominadas Redes Neuronales Recurrentes (RNN). Las RNN son redes especializadas en el procesamiento de secuencias, manteniendo un flujo recurrente de información en la red. Al poder mantener información de los estados anteriores de la red, las RNN realizan predicciones de mejor calidad a un mayor horizonte de predicción [23].

Entre las RNN del estado del arte se encuentran la Long Short Term Memory (LSTM) y la Echo-State Network (ESN), ambas RNN manejan el flujo recurrente de distinta manera para realizar la predicción.

Para la predicción de potencia eólica, los modelos LSTM no cuentan con mucho desarrollo, en cambio los trabajos relacionados a la ESN se centran mayormente en desarrollar metodologías para la búsqueda de hiper-parámetros para el modelo. En [24] se muestran los hiper-parámetros más relevantes y proponen que la búsqueda de parámetros se enfatice en algunos más que otros. En [10, 22] utilizan algoritmos genéticos, un algoritmo heurístico que utiliza una población de modelos para buscar los mejores hiper-parámetros de la ESN. Otros trabajos [13] además de optimizar los hiper-parámetros del modelo, buscan la mejor arquitectura o cantidad de nodos de la ESN.

3 | Análisis de Series de Tiempo.

3.1. Proceso Estocástico

Un proceso estocástico es una familia de variables aleatorias $\{X_t\}_{t\in T}$ definidas sobre un espacio de probabilidad (Ω, Σ, P), donde Ω es un espacio de muestreo, Σ es un $\sigma - algebra$, por ejemplo una familia de eventos de Σ y P es una medida de probabilidad, indexadas por un conjunto T discreto o continuo.

Una representación de un proceso estocástico $X(t, w), t \in T, w \in \Omega$ está dada por la siguiente función:

$$X : T \times \Omega \rightarrow \mathbb{R} \tag{3.1}$$

$$(t, w) \rightarrow X(t, w) \tag{3.2}$$

donde para un tiempo $t_0 \in T$ fijo, se tiene una familia de variables aleatorias $X(w) \;\; w \in \Omega$, en cambio para un $w_0 \in \Omega$ fijo, se tiene una función de $t \in T$ denominada trayectoria del proceso estocástico.

Un proceso estocástico es distinguido por los valores que pueden tomar la familia de variables aleatorias $\{X_t\}_{t\in T}$ o espacio de estados, por su conjunto de índices T y por las relaciones de dependencia entre las variables aleatorias X_t. De acuerdo al teorema de extensión de Kolmogorov, conocemos la estructura probabilística del proceso estocástico cuando conocemos todas las distribuciones finito dimensionales [27].

Una característica relevante al realizar el estudio de un proceso estocástico es la estacionariedad de éste, la cual está asociada a la regularidad de sus características estadísticas en el conjunto indexado $t \in T$. La estacionariedad de un proceso se puede dividir en estacionariedad estricta o débil.

Dado un proceso estocástico $\{X_t\}_{t\in T}$ y una función de probabilidad conjunta finito dimensional F, se dice que el proceso es estrictamente estacionario si se cumple

$$F(X_{t_1}, \ldots, X_{t_k}) = F(X_{t_{1+h}}, \ldots, X_{t_{k+h}}) \tag{3.3}$$

es decir la distribución conjunta finito dimensional se mantiene invariante al desplazamiento del índice t.

Por otro lado un proceso estocástico $\{X_t\}_{t\in T}$ se denomina débilmente estacionario si la media y covarianza entre X_t y X_{t+h} es invariante en $t \in T$, donde h es un número entero arbitrario. Este tipo de proceso también es conocido como un proceso estacionario de segundo orden, esto quiere decir que su segundo momento es finito $\mathbb{E}[X_t^2] < \infty$. Formalmente un proceso estocástico $\{X_t\}_{t\in T}$ es débilmente estacionario si:

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

Donde la función $\gamma_X(\cdot)$ es llamada la función de autocovarianza de $\{X_t\}_{t\in T}$.

Un proceso estacionario utilizado para el modelamiento, es el conocido como ruido blanco. Un proceso estacionario $\{w_t\}_{t\in T}$ se denomina de ruido blanco si:

- $\mathbb{E}[w_t] = 0, \forall t \in \mathbb{T}$
- $\mathbb{E}[w_t^2] = \sigma^2, \forall t \in \mathbb{T}$
- $Cov(w_t, w_{t+h}) = 0, \forall t \in \mathbb{T}, h \in \mathbb{Z}, h \neq 0$

Dado que la covarianza entre w_t y w_{t+h} es igual a cero, no existe correlación entre las variables aleatorias del proceso. Además existe un proceso denominado ruido blanco fuerte, en el cual las variables aleatorias a diferentes $t \in T$ son Independientes e Idénticamente Distribuidas (IID). Uno de los procesos de ruido blanco más utilizados es el denominado ruido blanco gaussiano, el cual sigue una ley de distribución de probabilidad normal $w_t \sim \mathcal{N}(\mu, \sigma^2)$.

3.2. Series de Tiempo

Una serie de tiempo corresponde a una realización o trayectoria de un proceso estocástico $\{x_t\}_{t\in T}$, donde el conjunto de índices T corresponde al tiempo, el cual puede ser discreto o continuo. En general si se tiene un proceso discreto el conjunto T es un intervalo en $\mathbb{N}_0$, en cambio un proceso continuo el conjunto T es un intervalo en $\mathbb{R}_0^+$.

En la Figura 3.1 se muestra la serie de tiempo de un proceso de ruido blanco gaussiano en tiempo discreto.

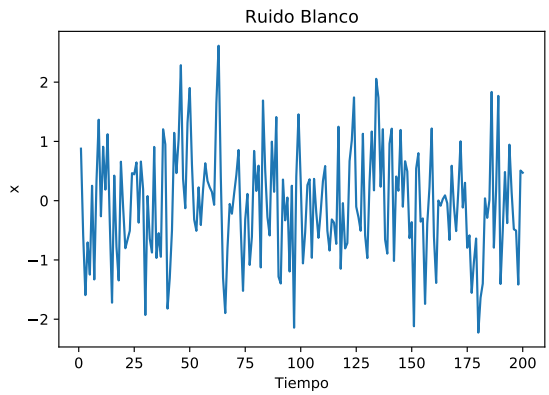

Figura 3.1: Ruido Blanco Gaussiano $x_t \sim \mathcal{N}(\mu, \sigma^2)$

Al momento de trabajar con series de tiempo se debe tener en consideración que las observaciones realizadas serán en tiempo discreto y además sólo se tendrá parte de la trayectoria del proceso, por lo que al momento de modelar el proceso es necesario realizar algunos supuestos como son el de estacionalidad débil y el de ergodicidad del proceso.

Este último supuesto está relacionado a que se tiene sólo una trayectoria finita del proceso, por lo que se supone que la función de medias y varianza del proceso tienden al valor real que es desconocido a medida que el tamaño de la serie de tiempo aumenta. Formalmente se dice un proceso estocástico $\{x_t\}_{t\in T}$ es ergódico si:

$$\mathbb{E}[x_t] = \begin{cases} \frac{1}{n}\sum_{i=i}^{n} x_{t_i} & \text{para un proceso de tiempo discreto.} \\ \frac{1}{T}\int_0^T x_t dt & \text{para un proceso de tiempo continuo.} \end{cases} \tag{3.4}$$

3.3. Modelo ARIMA y SARIMA

El modelo **Autoregresivo Integrado de Medias Móviles** (ARIMA) es un modelo lineal estocástico donde el proceso estocástico es modelado en término de sus valores pasados y la agregación de innovaciones aleatorias (schocks).

Dado el proceso estocástico $\{x_t\}_{t\in T}$ con media $\mu = 0$, el modelo $ARIMA(p, d, q)$ se define como

$$(1 - \phi_1 B - \phi_2 B^2 \cdots - \phi_p B^p)(1 - B^d)x_t = (1 + \theta_1 B + \theta_2 B^2 \cdots + \theta_q B^q)w_t, \tag{3.5}$$

Donde $\{w_t\}_{t\in T}$ es un proceso de ruido blanco con $\mathbb{E}[w_t] = 0$ y varianza σ^2, B es el operador de rezago tal que $B^k x_t = x_{t-k}$, p es el orden del operador autoregresivo $\Phi_p(B) = (1 - \phi_1 B - \phi_2 B^2 \cdots - \phi_p B^p)$, q es el orden del operador de medias móviles $\Theta_q(B) = (1 + \theta_1 B + \theta_2 B^2 \cdots + \theta_q B^q)$ y d es el orden de la derivada discreta $\Delta^d x_t = x_t - x_{t-d} = (1 - B^d)x_t$. Por simplicidad el modelo es representado de la forma

$$\Phi_p(B)\Delta^d x_t = \Theta_q(B)w_t, \tag{3.6}$$

El modelo supone que el proceso $\tilde{x}_t = \Delta^d x_t$ sea estacionario, utiliza el operador autoregresivo $\Phi_p(B)$ para incorporar la dependencia con valores pasados y el operador de medias móviles $\Theta_q(B)$ para incoporar el componente innovativo del proceso de ruido blanco $\{w_t\}_{t\in T}$.

Como se supone $\tilde{x}_t$ estacionario en realidad el modelo ajustado es un modelo del tipo $ARMA(p, q)$ de la serie transformada, en el caso de que $q = 0$ entonces tenemos un modelo autoregresivo de orden $AR(p)$, en cambio cuando $p = 0$ se tiene un modelo de medias móviles de orden $MA(q)$.

Al momento de ajustar los parámetros en el modelo hay que tener en cuenta dos propiedades que se tienen que cumplir, la **causalidad** e **invertibilidad** del proceso.

Un modelo $ARMA(p, q)$, $\Phi_p(B)\tilde{x}_t = \Theta_q(B)w_t$ se dice causal si la serie de tiempo $\{\tilde{x}_t; t = 0, \pm 1, \pm 2, \dots\}$ se puede escribir como un proceso lineal de la forma

$$\Psi(B)\tilde{x}_t = \sum_{j=0}^{\infty} \psi_j w_{t-j} \tag{3.7}$$

donde $\Psi(B) = \sum_{j=0}^{\infty} B^j$, $\sum_{j=0}^{\infty} |\psi_j| < \infty$ y $\psi_0 = 1$. Por lo que el proceso $ARMA(p,q)$ se puede expresar

$$\Phi_p(B)\Psi(B)w_t = \Theta_q(B)w_t \tag{3.8}$$

Esto nos dice que la estabilidad del operador $\Psi(B)$ en el plano complejo ($z \in \mathbb{C}$) está dada por

$$\Psi(z) = \sum_{j=0}^{\infty} \psi_j z^j = \frac{\Theta_q(z)}{\Phi_p(z)}, |z| \leq 1. \tag{3.9}$$

por lo que se tiene que cumplir que $\Phi_p(z) \neq 0$, esto quiere decir que las raíces del operador $\Phi_p(z)$ deben encontrarse fuera del circulo unitario en el plano complejo $|z| \leq 1$.

Por otro lado un modelo $ARMA(p,q)$ se dice que es invertible si la serie de tiempo $\{\tilde{x}_t; t = 0, \pm 1, \pm 2, \dots\}$ puede ser representada como un proceso autoregresivo infinito $AR(\infty)$ de la forma

$$\pi(B)\tilde{x}_t = \sum_{j=0}^{\infty} \pi_j \tilde{x}_{t-j} = w_t \tag{3.10}$$

donde $\pi(B) = \sum_{j=0}^{\infty} \pi_j B^j$, $\sum_{j=0}^{\infty} |\pi_j| < \infty$ y $\pi_0 = 1$. Reemplazando en la ecuación 3,6 se obtiene

$$\Phi_p(B)\tilde{x}_t = \Theta_q(B)\pi(B)\tilde{x}_t \tag{3.11}$$

Por lo que la estabilidad del operador $\pi(B)$ en el plano complejo está dada por

$$\pi(z) = \sum_{j=0}^{\infty} \pi_j z^j = \frac{\Phi_p(z)}{\Theta_q(z)}, |z| < 1 \tag{3.12}$$

análogo a la propiedad anterior se tiene que cumplir que $\Theta_q(z) \neq 0$, esto quiere decir que el proceso $ARMA(p,q)$ es invertible sólo si las raíces del operador $\Theta_q(z)$ se encuentran fuera del circulo unitario en el plano complejo. La propiedad de invertibilidad nos garantiza la unicidad en la determinación del modelo.

Las propiedades anteriormente mencionados suponen que la serie de tiempo $\{\tilde{x}_t; t = 0, \pm 1, \pm 2, \dots\}$ sea estacionaria, pero existen casos que este supuesto no se cumple. Cuando las series de tiempo presentan estacionalidad o comportaminetos cíclicos, ya sea como el comportamiento de las ventas de multitiendas en Navidad o San Valentín; como también por sucesos naturales como la velocidad

del viento el cual varia con cierta regularidad si es de día o noche, se utilizan modelos del tipo Seasonal ARIMA (SARIMA) que pueden ser multiplicativos o aditivos. Estos modelos dividen el proceso estocástico en dos componentes, una componente no-estacional y otra estacional que son modeladas de forma separada y luego integradas en un sólo modelo.

Dado el proceso estocástico $\{x_t\}_{t\in T}$ un modelo $SARIMA$ multiplicativo de orden $SARIMA(p,d,q)\times(P,D,Q)_S$ es de la forma

(3.13)

Donde $\Phi_p(B)$, $\Theta_q(B)$ son los operadores de orden p,q del componente no-estacional del proceso, Δ^d es la derivada del componente no-estacional, S es la frecuencia del componente estacional, $\Phi_P^S(B) = (1-\phi_1 B^S \ldots \phi_P B^{PS})$, $\Theta_Q^S(B) = (1+\theta_1 B^S \cdots + \theta_Q B^{QS})$ son los operadores autoregresivos y de medias móviles del componente estacional respectivamente, y $\Delta^{D\cdot S}$ es la derivada del componente estacional.

La metodología utilizada para el ajuste de un modelo $ARIMA(p,d,q)$ o $SARIMA(p,d,q)\times(P,D,Q)$ es similar, se comienza realizando una selección de los parámetros estructurales p,q en el caso de los modelos no estacionales y p,q,P,Q para los modelos estacionales. Luego de la selección de parámetros estructurales se realiza el ajuste del modelo a los datos, para esto existen distintos métodos para estimar los parámetros del modelo, luego de estimar los parámetros del modelo se puede realizar predicciones a distintos horizontes de predicción.

3.3.1. Selección de los parámetros estructurales

Para seleccionar el orden p,q del modelo $ARMA(p,q)$ que se desea ajustar a una serie de tiempo proveniente de un proceso estacionario, es conveniente realizar un análisis de las funciones de autocorrelación (ACF) y autocorrelación parcial de la serie de tiempo observada.

La función de autocorrelación empírica de una serie de tiempo $\{x_t; t = 0, \pm1, \pm2, \ldots\}$ está dada por

$$\hat{\rho}_h = \frac{\hat{\gamma}_{x_t}(h)}{\hat{\gamma}_{x_t}(0)} \tag{3.14}$$

con $\hat{\gamma}_{x_t}(h)$, la función de autocovarianza empírica definida como

$$\hat{\gamma}_{x_t}(h) = \frac{1}{n}\sum_{t=1}^{n-h}(x_{t+h} - \overline{x})(x_t - \overline{x}), \tag{3.15}$$

donde n es la cantidad de observaciones de la serie de tiempo, $\overline{x}$ es la media de la serie de tiempo. Cabe mencionar que $\hat{\gamma}_{x_t}(h) = \hat{\gamma}_{x_t}(-h)$.

La función de autocorrelación permite estimar la dependencia lineal entre dos instantes de la serie de tiempo separados por h rezagos o *lags*. Por otro lado la función de autocorrelación parcial nos entrega la correlación entre dos instantes sin tomar en cuenta la influencia de los instantes intermedios.

La función de autocorrelación parcial (PACF) se puede estimar mediante las ecuaciones de Yule-Walker a partir de los coeficientes ACF, de modo que el coeficiente de h rezagos $\hat{\alpha}_h$ se puede estimar como

$$\hat{\alpha}_1 = \hat{\rho}_1, \qquad \hat{\alpha}_2 = \frac{\begin{vmatrix} 1 & \hat{\rho}_1 \\ \hat{\rho}_1 & \hat{\rho}_2 \end{vmatrix}}{\begin{vmatrix} 1 & \hat{\rho}_1 \\ \hat{\rho}_1 & 1 \end{vmatrix}}, \qquad \hat{\alpha}_3 = \frac{\begin{vmatrix} 1 & \hat{\rho}_1 & \hat{\rho}_1 \\ \hat{\rho}_1 & 1 & \hat{\rho}_2 \\ \hat{\rho}_2 & \hat{\rho}_1 & \hat{\rho}_3 \end{vmatrix}}{\begin{vmatrix} 1 & \hat{\rho}_1 & \hat{\rho}_1 \\ \hat{\rho}_1 & 1 & \hat{\rho}_2 \\ \hat{\rho}_2 & \hat{\rho}_1 & 1 \end{vmatrix}} \tag{3.16}$$

y así sucesivamente hasta llegar al coeficiente $\hat{\alpha}_h$.

Luego de estimar los coeficientes de la función de autocorrelación y autocorrelación parcial de la serie de tiempo analizada a distintos rezagos $1, \ldots, h$, estos posteriormente son graficados. Dependiendo del comportamiento de la gráfica de las funciones se puede distinguir el tipo de proceso de donde proviene la serie de de tiempo, que puede ser un proceso $AR(p)$, $MA(q)$ o $ARMA(p, q)$ según el comportamiento de los grafícos de ACF y PACF (tabla 3.1).

En el caso de series estacionales, al momento de realizar el análisis de los gráficos de ACF y PACF se puede observar que cada S rezagos existe un aumento en el coeficiente de estas funciones, donde S es la frecuencia del componente estacional del proceso.

Modelo	**ACF**	**PACF**
AR(p)	Decrecimiento rápido exponencial u ondas sinusoidales	Sólo los primeros p coeficientes son significativos. El resto se anulan bruscamente.
MA(q)	Sólo los primeros q coeficientes son significativos. El resto se anulan bruscamente.	Decrecimiento rápido exponencial u ondas sinusoidales
ARMA(p,q)	Decrecimiento rápido exponencial u ondas sinusoidales	Decrecimiento rápido exponencial u ondas sinusoidales

Tabla 3.1: Modelo subyacente según el comportamiento de los gráficos de ACF y PACF a distintos rezagos.

Un método alternativo para seleccionar los parámetros estructurales del modelo $ARMA(p,q)$ es mediante el uso del criterio de información de Akaike (AIC). El criterio de información de Akaike (AIC) se define como

$$AIC = \ln \hat{\sigma}_k^2 + \frac{n+2k}{n} \tag{3.17}$$

donde $\hat{\sigma}_k^2$ es el estimador de máxima verosimilitud de la varianza de la innovaciones del modelo $ARMA(p,q)$, $k = p + q$ es el orden del modelo y n es la cantidad de observaciones de la serie de tiempo. El criterio AIC toma en consideración el ajuste del modelo y también la cantidad de parámetros del modelo.

3.3.2. Estimación de parámetros

Luego de seleccionar los parámetros estructurales p, q del modelo $ARMA(p,q)$ es necesario ajustar los parámetros $\beta = \{\phi_1, \phi_2, \ldots, \phi_p, \theta_1, \theta_2, \ldots, \theta_q\}$ a la serie de tiempo. Existen infinitos modelos $ARMA(p,q)$, por lo que es necesario utilizar alguna técnica para encontrar el modelo óptimo, por ejemplo el criterio de optimización de máxima verosimilitud (MLE).

Dada una serie de tiempo $\{x_t; t = 0, \pm 1, \pm 2, \ldots\}$ proveniente de un proceso estacionario, el estimador de máxima verosimilitud busca los parámetros $\beta = \{\phi_1, \phi_2, \ldots, \phi_p, \theta_1, \theta_2, \ldots, \theta_q\}$ que maximizan la función de distribución conjunta

$$f(x;\beta) = f(x_0, x_2, \ldots x_n; \beta) = \mathcal{L}(\beta|x) \tag{3.18}$$

Donde $\mathcal{L}(\beta|x)$ es la función de verosimilitud de los parámetros β con respecto a la serie de tiempo $x = \{x_t\}$. Suponiendo que el proceso asociado a la serie de tiempo $\{x_t\}$ es Gaussiano, la función de verosimilitud puede ser expresada de la forma

$$\mathcal{L}(\beta|x) = f(x;\beta) = \left(\prod_{t=p+1}^{n} f(x_t|I_{t-1},\beta)\right) f(x_p,\ldots,x_1;\beta) \tag{3.19}$$

con $I_{t-1} = x_t,\ldots,x_1$ la información disponible al instante t, y $x_p,\ldots,x_1$ denota los valores iniciales. A su vez la función de log-verosimilitud se define como

$$\ln \mathcal{L}(\beta|x) = \sum_{t=p+1}^{n} \ln f(x_t|I_{t-1},\beta) + \ln f(x_p,\ldots,x_1;\beta) \tag{3.20}$$

El estimador de máxima verosimilitud busca

$$\hat{\beta}_{MLE} = \arg\max_{\beta} \sum_{t=p+1}^{n} \ln f(x_t|I_{t-1},\beta) + \ln f(x_p,\ldots,x_1;\beta) \tag{3.21}$$

Al momento de calcular el estimador $\hat{\beta}_{MLE}$ hay que tener en cuenta las propiedades de causalidad e invertibilidad del modelo $ARMA(p,q)$. En el caso del modelo $ARMA(p,q)$ Gaussiano, el estimador de máxima verosimilitud no tiene una solución exacta por lo que se recurre a aproximaciones numéricas para obtener los valores de $\hat{\beta}_{MLE}$, un ejemplo es el uso de un algoritmo del tipo Newton Raphson, un algoritmo iterativo que se define como

$$\hat{\beta}_{MLE,i} = \hat{\beta}_{MLE,i-1} - \hat{H}(\hat{\beta}_{MLE,i-1})^{-1}\hat{s}(\hat{\beta}_{MLE,i-1}) \tag{3.22}$$

donde $\hat{H}(\hat{\beta})$ es la segunda derivada de la función de log-verosimilitud o matriz Hessiana, $\hat{s}(\hat{\beta})$ la primera derivada de la log-verosimilitud, i es la iteración actual del algoritmo. La estimación de las derivadas puede ser realizada mediante métodos numéricos de derivación o pueden ser calculadas de forma analítica si es posible.

3.3.3. Predicción del modelo

Luego de estimar los parámetros del modelo es posible realizar predicciones de valores futuros de la serie x_{n+h} con h el horizonte de predicción. Suponiendo que el proceso $ARMA(p, q)$ modelado sigue una distribución Gaussiana con media cero y varianza σ^2, se puede asumir que los valores futuros también tienen la misma distribución. Por esto último el valor esperado del modelo ARMA corresponde a la estimación puntual del modelo

$$\hat{x}_{n+h} = \sum_{j=1}^{p} \phi_j x_{n+(h-j)} + \sum_{k=1}^{q} \theta_k w_{n+h-j} \tag{3.23}$$

Debido a que se está realizando una estimación puntual de un proceso estocástico existe un grado de incertidumbre en esa predicción, por lo mismo es común utilizar intervalos de confianza para manejar dicha incertidumbre. Los intervalos de confianza entregan con un nivel de confianza $100(1 - \alpha)\,\%$ que el valor futuro se encuentra dentro del intervalo.

El cálculo de un intervalo con confianza α de un modelo ARMA se define como

$$\hat{x}_{t+h} \pm Z_{1-\frac{\alpha}{2}} \sqrt{\hat{\sigma}_w^2 \sum_{j=0}^{h-1} \psi_j^2} \tag{3.24}$$

donde $Z_{1-\frac{\alpha}{2}}$ es el cuantil $1 - \frac{\alpha}{2}$ de una distribución normal estándar, ψ_j son los coeficientes del operador utilizado para definir la propiedad de causalidad en la ecuación 3.9 y $\hat{\sigma}_w^2$ es la estimación de la varianza del ruido blanco del proceso.

4 Redes Neuronales Recurrentes.

Las Redes Neuronales Recurrentes (RNN) son una familia de las Redes Neuronales Artificiales (ANN) que se especializa en el procesamiento de datos secuenciales, por lo que son útiles en problemas donde existe dependencia temporal por ejemplo, procesamiento vídeo y series de tiempo.

Una RNN se puede ver como un sistema dinámico de la forma

$$y^{(t)}, s^{(t)} = g(x^{(t)}, s^{(t-1)}) \tag{4.1}$$

donde $x^{(t)} = (x_1^{(t)}, x_2^{(t)}, \ldots, x_I^{(t)})^T$ es el vector de entrada de la secuencia en el instante t, $y^{(t)} = (y_1^{(t)}, y_2^{(t)}, \ldots, y_O^{(t)},)^T$ es el vector de salida en el instante t, $s^{(t)}$ es el estado interno del modelo y $g(\cdot)$ es la función de transición que realiza el mapeo desde el instante anterior al actual. Este sistema dinámico puede ser representado como un grafo recursivo (Figura 4.1).

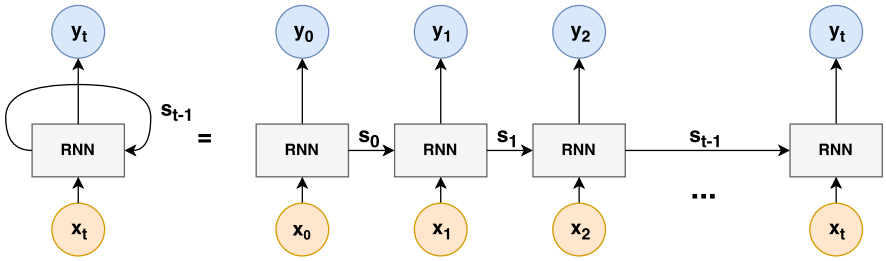

Figura 4.1: Desarrollo de la RNN recursivamente llegando al instante inicial

Las RNN utilizan el mismo modelo para realizar la predicción de la secuencia a distintos instantes t, por lo que se dice que los parámetros entre los distintos instantes de la secuencia. Debido a esta propiedad pueden ser utilizadas para procesar secuencias de largo variable.

Dependiendo de la arquitectura de la RNN la función de transición $g(\cdot)$ puede variar, en términos generales la función de transición se puede representar como una grafo de tres capas (Figura 4.2), capa de entrada, capa oculta y capa de salida.

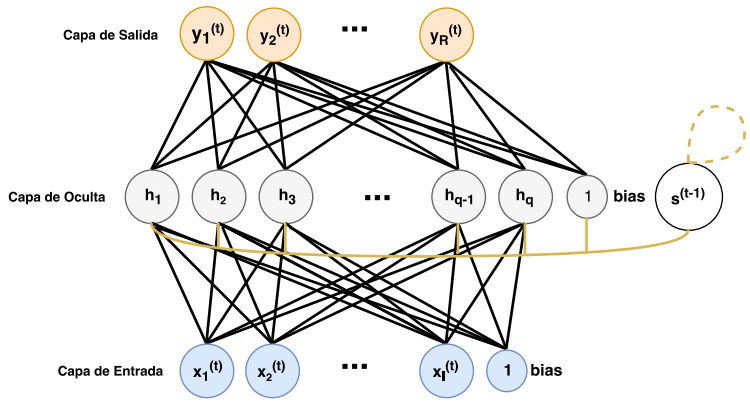

Figura 4.2: RNN Genérica, dependiendo de la arquitectura el flujo de información a los estados $s^{(t)}$ puede variar.

Los nodos de la capa de entrada representan a cada valor o *feature* del vector de entrada $x^{(t)}$. Por su parte los nodos de las otras capas toman como entrada los valores provenientes de la capa anterior y entregan un valor de salida utilizando una combinación lineal de la forma

$$a_{q,d} = f\left(w_0 + \sum_{j=1}^{m} w_{qj} a_{j,d-1}\right) \tag{4.2}$$

donde $a_{q,d}$ es la salida del nodo q en la capa d, w_0 es el *bias* o sesgo, w_{qj} es el peso o conexión entre el $j-esimo$ nodo de la capa anterior y el nodo q de la capa d, y $f(\cdot)$ es una función que puede ser no lineal denominada función de activación.

La función de activación de una RNN es la encargada de otorgar la capacidad de encontrar relaciones no lineales entre los valores de entrada. Entre las funciones de activación más utilizadas se encuentran, la tangente hiperbólica (*tanh*) definida como

$$\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}, \tag{4.3}$$

la función sigmoidal (σ) definida como

$$\sigma(z) = \frac{1}{1 + e^{-t}} \tag{4.4}$$

y la Rectified Unified Unit (ReLU) que se define como

$$ReLU(z) = z^{+} = max(0, z). \tag{4.5}$$

Las funciones *tanh* y sigmoidal se caracterizan por realizar una transición suave entre $[-1, 1]$ y $[0, 1]$ respectivamente, en cambio la función ReLU no está acotada y su transición no es suave por lo que no existe una derivada cuando $z = 0$ (Figura 4.3).

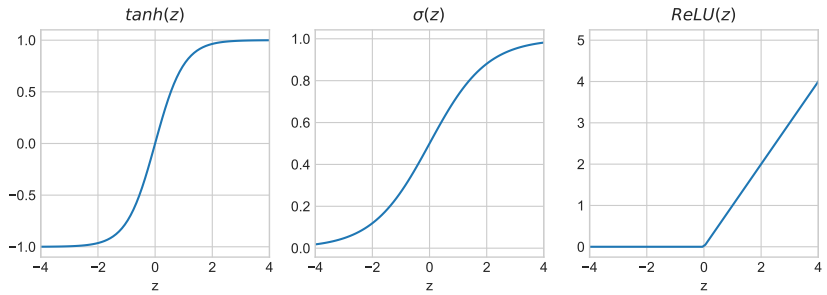

Figura 4.3: Gráfica de funciones de activación

Luego de definir la arquitectura y función de activación, es necesario buscar los parámetros del modelo w_{qj} que mejor ajustan a los datos. Existen diversas técnicas de realizar esta optimización, en general lo que se busca son los parámetros w_{qj} que minimizan una función de pérdida L. En los problemas de predicción se puede utilizar como función de perdida el error cuadrático medio (MSE), definido por

$$L = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{4.6}$$

donde y_i es el valor real de la salida de la $i - esima$ secuencia, $\hat{y}_i = g(x_i^{(t)}, s^{(t-1)})$ es el valor de salida de la RNN luego procesar la $i - esima$ secuencia hasta el instante t. El proceso de buscar los parámetros que minimizan la función de pérdida también se denomina entrenamiento de la red, ya que se asocia a enseñar al modelo a procesar los datos.

Al momento de trabajar con RNN, primero es necesario seleccionar el tipo o arquitectura de RNN que se utilizará, ya que la función de transición y el manejo del estado interno de la red dependerá de la arquitectura de la RNN. Luego de seleccionar la arquitectura de la RNN se recomienza realizar un procesamiento previo de los datos, preparando los datos para el modelo pueda utilizarlos eficazmente. Luego de preprocesar los datos se selecciona el mejor modelo según una medida de desempeño.

4.1. Arquitecturas de RNN

Unas de las primeras arquitecturas de RNN fue propuesta por Hopfield [18], la red de Hopfield tiene nodos binarios, y se utiliza para almacenar patrones. El estado de un nodo en una red de Hopfield está influenciado por la conexiones recurrentes con los otros nodos (Figura 4.4), al ingresar un nuevo patrón, mediante una regla de actualización de los nodos, la red se estabiliza en uno de los patrones almacenados por la red.

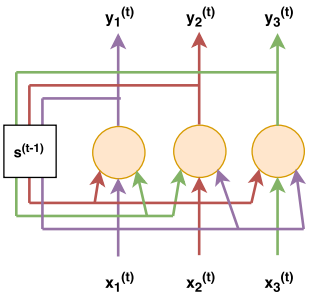

Figura 4.4: Red Hopfield con tres nodos.

Otras arquitecturas de RNN están pensadas para problemas supervisados, es decir, se conoce la secuencia de entrada $\{x^{(0)}, x^{(1)}, \ldots x^{(t)}\}$ y el vector de salida $y^{(t)}$. Entre las RNN utilizadas para problemas supervisados se encuentra la propuesta por Jordan (1986).

La red de Jordan extiende la capa oculta con unas unidades especiales denominadas *Context Units*, el valor de salida de la red en el instante $t-1$ es ingresado a estas unidades y luego son ingresadas a la capa oculta en el instante siguiente $s^{(t-1)} = y^{(t-1)}$, formando una recurrencia (Figura 4.5).

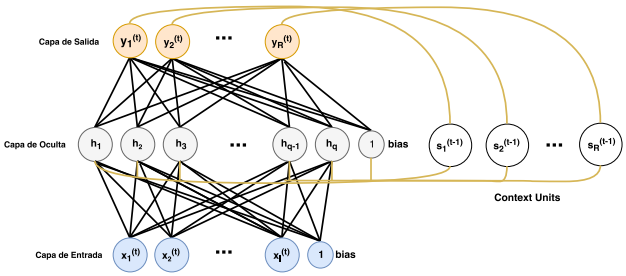

Figura 4.5: Red de Jordan

Una variación a la red de Jordan es la propuesta por Elman [12], en la red de Elman se ingresa la salida de los nodos de la capa oculta a las *Context Units*.

Para ajustar los pesos de RNN supervisadas, como las de Jordan y Elman, se utiliza una técnica de optimización iterativa denominada *Back Propagation Through Time* (BPTT).

El BPTT es un algoritmo de gradiente descendente donde las matrices de pesos W_{hx}, W_{hy}, U entre la capa de entrada y oculta, la capa oculta y de salida y la recurrencia se actualizan mediante la siguiente ecuación

$$W^{i+1} = W^{i} - \alpha \nabla_W L(y^{(t)}, \hat{y}^{(t)})$$

Donde W^i es una de la matrices de pesos en la iteración i, α es un valor entre $[0, 1]$ llamado *learning rate* y $\nabla_W L(y^{(t)}, \hat{y^{(t)}})$ es el gradiente de la función de pérdida en una secuencia $(\{x^{(0)}, x^{(1)}, \ldots, x^{(t)}\}, y^{(t)})$.

Para calcular $L(y^{(t)}, \hat{y}^{(t)})$ se realiza un *forward-propagation through time* , por lo que se desarolla la RNN como en la Figura 4.1, se calcula la salida $y^{(t_0)}$ en cada instante $0 \leq t_0 \leq t$ utilizando la ecuación 4,2 y se actualiza el estado interno de la red $s^{(t_0)}$ dependiendo de la arquitectura utilizada.

Luego de realizar el *forward-propagation through time*, se calcula el gradiente $\nabla_W L(y^{(t)}, \hat{y}^{(t)})$ de W_{hk}. Este gradiente se utiliza para el cálculo del gradiente en la capa anterior. Paso denominado *Backpropagation*.

Como existe dependencia entre los distintos estados $s^{(t_0)}$ que va tomando la red, el error de los pesos recurrentes U, además se tiene que propagar a traves de los distintos instantes de tiempo, paso que se denomina *Backpropagation through time*.

La mayor dificultad al momento de utilizar redes de Jordan y Elman es la poca capacidad de aprender dependencias a largo plazo debido al *vanishing* y *exploding gradient* [2], que están asociados a la disminución rápida o explosiva del gradiente respectivamente cuando se propaga a través de las capas o los instantes de tiempo. Debido a esto han surgido nuevas arquitecturas pensadas en abordar esta problemática.

Entre las arquitecturas de RNN modernas más desarolladas se encuentran la **Long-short Term Memory** y la **Echo-State Network**.

4.1.1. Long Short Term Memory

Hochreiter y Shmidhuber proponen el modelo *Long Short Term Memory* (LSTM) [17] para superar el problema del *vanishing gradient* y así poder aprender dependencias a largo plazo. En este modelo se reemplaza los nodos de la capa oculta por unos nodos especiales denominados *memory cell* (Figura 4.6).

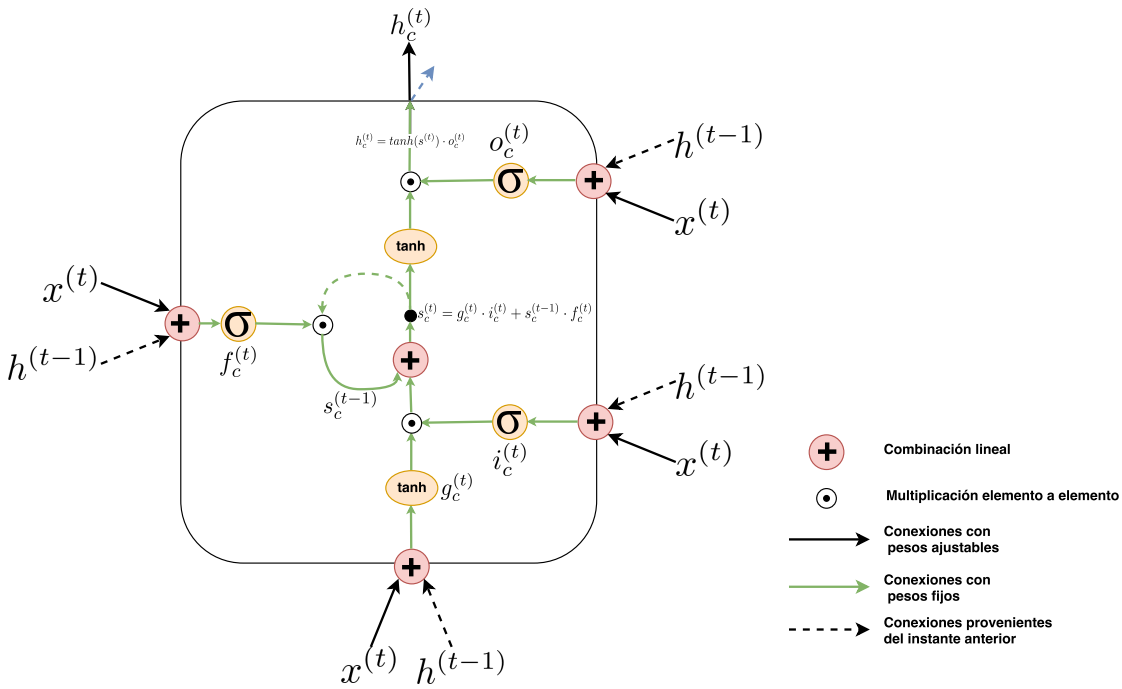

Figura 4.6: Una *memory cell* del modelo LSTM. El subíndice c denota el valor de la unidad, por lo que se puede expresar los valores de la capa de forma vectorial sin el subíndice.

Cada *memory cell* tiene una una conexión recurrente $s^{(t)}$ con un peso fijo , asegurando que el gradiente al propagarse a través del tiempo no disminuya rápidamente o explote.

Una *memory cell* está compuesta por nodos simples en un patrón de conexión especifico. Entre los nodos que componen una *memory cell* se encuentran una serie de compuertas que se encargan de manejar el flujo de información de la unidad. Los componentes de una *memory cell* se detallan a continuación

1. Input Node (g_c): El nodo de entrada realiza la combinación lineal del vector de entrada $x^{(t)}$ y la salida de la capa oculta en el instante anterior $h^{(t-1)}$. Este nodo entrega nueva información a la *memory cell*.

2. Input Gate (i_c): Es la primera compuerta utilizada por la LSTM, controla el flujo de información que entra a la *memory cell*, si el valor de la compuerta es cero, no ingresa nueva información a la *memory cell*, en cambio si es uno, toda la nueva información ingresa a la *memory cell*. El valor del *input gate* se mutliplica, por la salida del *input node*.

3. Internal State (s_c): El nodo principal del modelo LSTM es el estado interno s_c, que utiliza una función de activación lineal y pesos fijos. Como se utiliza un peso fijo en la recurrencia del estado interno, el error se propaga a traves del tiempo sin los problemas de *vanishing* y *exploding gradient*. Esta conexión recurrente también se denomina ***constant error carrousel***. La actualización del estado interno está dada en notación vectorial por $s^{(t)} = g^{(t)} \odot i^{(t)} + s^{(t-1)}$, donde $\odot$ es la multiplicación elemento a elemento entre los vectores.

4. Forget Gate (f_c): Estas compuertas fueron introducidas por [14]. Permiten restringir la información que se mantiene en el estado interno de la *memory cell*. Cuando se utiliza una *forget gate* la actualización del estado interno se define como

$$s^{(t)} = g^{(t)} \odot i^{(t)} + f^{(t)} \odot s^{(t-1)} \tag{4.7}$$

5. Output Gate (o_c): El *output gate* controla el flujo de salida de información de la *memory cell*. El valor que entrega finalmente una *memory cell*, está dado por el estado interno de la unidad y la *output gate*.

Dada una secuencia de entrada $\{x^{(0)}, x^{(1)}, \ldots, x^{(t)}\}$, y utilizando los componentes de la *memory cell*, el *forward-propagation* de una LSTM se define con las siguientes ecuaciones

$$g^{(t)} = tanh(W^{gx}x^{(t)} + W^{gh}h^{(t-1)} + b_g) \tag{4.8}$$

$$i^{(t)} = \sigma(W^{ix}x^{(t)} + W^{ih}h^{(t-1)} + b_i) \tag{4.9}$$

$$f^{(t)} = \sigma(W^{if}x^{(t)} + W^{if}h^{(t-1)} + b_f) \tag{4.10}$$

$$o^{(t)} = \sigma(W^{io}x^{(t)} + W^{io}h^{(t-1)} + b_f) \tag{4.11}$$

$$s^{(t)} = g^{(t)} \odot i^{(t)} + f^{(t)} \odot s^{(t-1)} \tag{4.12}$$

$$h^{(t)} = tanh(s^{(t)}) \odot o^{(t)} \tag{4.13}$$

donde t es el instante de la secuencia, $t-1$ el instante anterior, $h^{(t-1)}$ son los valores de salida de la capa oculta de la red en el instante anterior, W^{ab} es la matriz de pesos que conecta la capa b con la capa a, b_a es el *bias* o sesgo de la capa a, *tanh* y σ son las funciones de activación propuestas por los autores,estas pueden variar.

Lo que se busca en la LSTM es aprender cuando dejar ingresar nueva información al estado interno y cuando dejar salir la información de la *memory cell*. Para el aprendizaje se utiliza BPTT, ajustando los pesos de las compuertas y de los nodos de entrada y salida.

En la práctica, el modelo LSTM tiene una mayor habilidad para aprender dependencias a largo plazo en comparación a las RNN simples.

4.1.2. Echo State Network

La Echo State Network (ESN) propuesta en [19] busca evitar los problemas del BPTT fijando los valores de las conexiones recurrentes de tal forma de que sean buenos capturando la historia del pasado, y sólo aprende los pesos de salida del modelo.

Este modelo pertenece al área denominada *Reservoir Computing* (RC), debido a que los nodos de la capa oculta crean un reservorio de características temporales, que pueden capturar diferentes relaciones temporales de los valores ingresados al modelo.

Una ESN está compuesta por tres capas, con la particularidad que la capa oculta, llamada reservorio, contienen una gran cantidad de nodos con conexiones recurrentes entre ellos (Figura 4.7).El reservorio se utiliza para dos propósitos, dado un vector de entrada de tamaño I y un reservorio de tamaño N, el reservorio realiza una expansión no lineal $\mathbb{R}^I \rightarrow \mathbb{R}^N$. Además el repositorio es el encargado de mantener la memoria del modelo, logrado gracias a las conexiones recurrentes $s^{(t-1)} = h^{(t-1)}$.

Para inicializar los pesos entre la capa de entrada y el reservorio W_{in}, y los pesos de las conexiones recurrentes entre los nodos del reservorio W se utilizan valores aleatorios, generalmente tomados de una distribución uniforme entre $[-1, 1]$. Usualmente los pesos recurrentes del reservorio W son una matriz *sparse*, eso quiere decir que gran cantidad de ellos son igual a cero, asegurando que la información de los estados anteriores provenga de distintas fuentes en cada nodo.

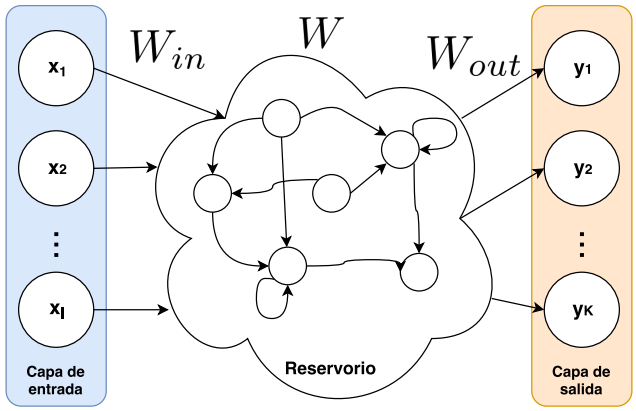

Figura 4.7: Echo State Network, la arquitectura está dividida en tres capas, una capa de entrada, un reservorio y una capa de salida.

Además de la propiedad *sparse* de los pesos recurrentes W, se tiene que cumplir el **estado echo**, el estado echo sólo se cumple si el estado actual $s^{(t)}$ es determinado únicamente por los valores anteriores de la secuencia $\{x^{(t)}, x^{(t-1)}, x^{(t-2)}, \dots\}$. Esto quiere decir que no hay dos secuencias diferentes que entreguen el mismo estado interno de la ESN.

En la práctica para cumplir el estado echo en la mayoría de los casos, se tiene que cumplir que el **radio espectral** $|\rho(W)| \leq 1$. Donde el radio espectral está dado por el mayor valor propio $|\lambda_{max}|$ de la matriz W.

Para ajustar los pesos de salida W_{out}, se comienza recolectando los posibles estados del reservorio realizando un *forward-propagation* de todas las secuencias. Dado M secuencias $\{x_i^{(0)}, x_i^{(1)}, \dots, x_i^{(t)}\}$

$$s_i^{(t)} = (1-a)s^{(t-1)} + a \cdot f(W^{in}x_i^{(t)} + Ws_i^{(t-1)}) \ \ , i = 1 \dots M \tag{4.14}$$

donde $s_i^{(t)} = h_i^{(t)}$ es el estado interno de la ESN al instante t de la $i-esima$ secuencia, f es una función de activación (Figura 4.3), y $x_i^{(t)}$ es el valor de entrada en el instante t de la $i-esima$ secuencia, y $\alpha \in [0,1]$ es denominado *leaking rate*, que ayuda a controlar la información que se mantiene dentro del estado.

Al realizar el *forward-propagation* con todas las secuencias se obtiene una matriz de estados $S \in \mathbb{R}^{\mathbb{M}\times\mathbb{N}}$ que se concatena con la matriz de entradas $X^{(t)} \in \mathbb{R}^{M\times I}$ por lo que podemos buscar los pesos W_{out} que minimizan la ecuación

$$[X;S] \cdot W_{out} = Y \tag{4.15}$$

donde $[X; S] \in \mathbb{R}^{M\times(N+I)}$ es la matriz de entradas y estados finales de cada secuencia, e $Y \in \mathbb{R}^{\mathbb{M}\times\mathbb{R}}$ es la matriz de vectores de salida de las secuencias.

Utilizando una función de pérdida cuadrática como el MSE 4.6, la ecuación 4.15 se puede resolver mediante mínimos cuadrados o una versión regularizada como *Lasso* o *Ridge Regression*. La solución de la ecuación 4.15 con *Ridge Regression* se obtiene de

$$W_{out} = ([X; S]^T [X; S] + \alpha I)^{-1} [X; S] Y \tag{4.16}$$

donde α es el coeficiente de regularización para la *Ridge Regression*.

El *forward-propagation* con los pesos de salida W_{out}, para una nueva secuencia se calcula con las siguientes ecuaciones

$$s^{(t)} = (1 - a)s^{(t-1)} + a \cdot f(W^{in} x^{(t)} + W s^{(t-1)}) \tag{4.17}$$

$$y^{(t)} = W_{out}[x^{(t)}; s^{(t)}] \tag{4.18}$$

donde $[x^{(t)}; s^{(t)}]$ es el vector de entrada y el estado interno en el instante t, e $y^{(t)}$ es el vector de salida para el instante t.

Al no utilizar BPTT como algoritmo de entrenamiento se evitan los problemas del *vanishing* y *exploding gradient*, además en la práctica se logran resultados comparables a otras RNN en un tiempo de entrenamiento mucho menor.

4.2. Preprocesamiento

El preprocesamiento de datos es una técnica, que toma los datos sin procesar y los prepara para poder ser utilizados por el modelo. Este proceso puede ser dividido en tres etapas

1. **Limpieza de los datos:** Esta etapa se encarga cuando se tienen datos faltantes, ruido y *outliers* en los datos. Cuando existen datos faltantes se pueden eliminar o utilizar técnicas de inputación para agregar nuevos valores. En el caso de ruido y outliers su manejo dependerá del problema y el modelo que se utilice.

2. **Transformación de los datos:** Dependiendo del modelo utilizado, este puede ser sensible a las escalas de los datos o si se encuentran centrados, por lo que se realizan transformaciones para que el modelo pueda ser utilizado eficazmente. Las transformaciones habituales son la **Estandarización** y **Normalización** de los datos.

   - Estandarización: La estandarización supone que los datos tienen una distribución normal $x \sim \mathcal{N}(\mu, \sigma^2)$ y transforma a una distribución normal estándar $x_{new} \sim \mathcal{N}(0, 1)$ según la ecuación
   $$x_{new} = \frac{x - \mu}{\sigma^2}, \tag{4.19}$$
   donde μ es la media y σ^2 es la varianza de los datos. Centrando los datos en el origen cartesiano.

   - Normalización: La normalización de los datos los transforma a un rango definido $[a, b]$ según la ecuación
   $$x_{new} = \frac{x - min(x)}{max(x) - min(x)}(b - a) + a, \tag{4.20}$$
   de esta forma todos los datos se van a encontrar en la misma escala. Usualmente se utilizan los rangos $[0, 1]$ y $[-1, 1]$ con modelos RNN.

3. **Reducción de dimensionalidad:** La reducción de dimensionalidad es un proceso para reducir el número de variables que tienen los datos. Este proceso se pueden dividir en dos enfoques, **Selección de Características** y **Extracción de Características**.

   - Selección de Características: La selección de características trata de encontrar un subconjunto de las variables originales.

   - Extracción de Características: La extracción de características utiliza una función $h : \mathbb{R}^n \rightarrow \mathbb{R}^m$ que lleva los datos de una alta dimensionalidad a una menor $m < n$.

$$\begin{aligned} h &: \mathbb{R}^n \rightarrow \mathbb{R}^m \\ &: x \rightarrow h(x) \end{aligned} \tag{4.21}$$

en problemas de Series de Tiempo, este paso no se utiliza a no ser que se esté utilizando una serie multivariada con muchas variables.

4.3. Selección de Modelo

Cuando se utilizan modelos de RNN para la predicción, es necesario que el modelo se ajuste a los datos que se tienen y además que tenga la capacidad de generalizar, es decir obtener buenos resultados para datos no observados.

Usualmente para simular datos no observados se divide los datos en dos conjuntos, un **conjunto de entrenamiento** utilizado para ajustar los pesos del modelo, y un **conjunto de prueba** utilizado para medir qué tan bien generaliza el modelo. Al trabajar con series de tiempo, esta división se tiene que hacer de forma secuencial para mantener la estructura de dependencia de los datos.

Luego de dividir los datos, se seleccionan los **hiper-parámetros** de la RNN, estos pueden variar según la arquitectura, pero dentro de los más comunes son la selección de la función de activación y la cantidad de nodos de la capa oculta.

Una metodología que se puede utilizar al momento de seleccionar los hiper-parámetros , es dividir nuevamente el conjunto de entrenamiento en un conjunto para ajustar los parámetros y otro para validar el modelo, también tomando en cuenta la secuencialidad de los datos. Utilizando el nuevo conjunto de entrenamiento se entrenan distintos modelos variando los hiper-parámetros.

Para medir el desempeño de los modelos entrenados se necesita definir alguna métrica, las más utilizadas en problemas de predicción de potencia eólica son *Root Mean Squared Error* y *Mean Absolute Percentage Error*.

Dado un conjunto de datos de entrada $x_i = \{x^{(0)}, x^{(1)}, \ldots, x^{(t)}\}$ y su valor de salida $y_i = y^{(t)}$ el *Root Mean Squared Error* (RMSE) se define como

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}. \tag{4.22}$$

Por su parte el *Mean Absolute Percentage Error* (MAPE) definido por la ecuación

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{4.23}$$

donde n es la cantidad de secuencias del conjunto e $\hat{y}_i = g(x_i^{(t)}, s^{(t-1)})$ es el valor de salida de la RNN para la secuencia de entrada $x_i = \{x^{(0)}, x^{(1)}, \ldots, x^{(t)}\}$.

El MAPE al utilizar el valor absoluto es menos suceptible a los posibles outliers presentes en los datos o las predicciones.

Dentro de los modelos entrenados se selecciona el modelo con menor RMSE o MAPE en el conjunto de validación, que se entrena nuevamente utilizando todo el conjunto de entrenamiento, y se calcula la métrica de desempeño para el conjunto de pruebas.

5 | Caso de Estudio.

Este trabajo busca aplicar modelos de predicción estadísticos clásicos y de Redes Neuronales Recurrentes (RNN) a datos generados por las plantas eólicas Chilenas. En especifico en este trabajo se utilizaron los datos de potencia real reportados por tres plantas eólicas del CDEC. Estos datos son horarios y actualizan diariamente, abarcando desde el año 2008 hasta la actualidad en el caso de las plantas estudiadas.

Luego de procesar inicialmente los datos, descartando datos de plantas que no fuesen eólicas, se obtiene el gráfico de la Figura 5.1. De este gráfico se observa que inicialmente la potencia eólica que se introducía en el sistema era muy pequeña, pero con el paso del tiempo y las nuevas plantas que se construyeron esta fue creciendo rápidamente.

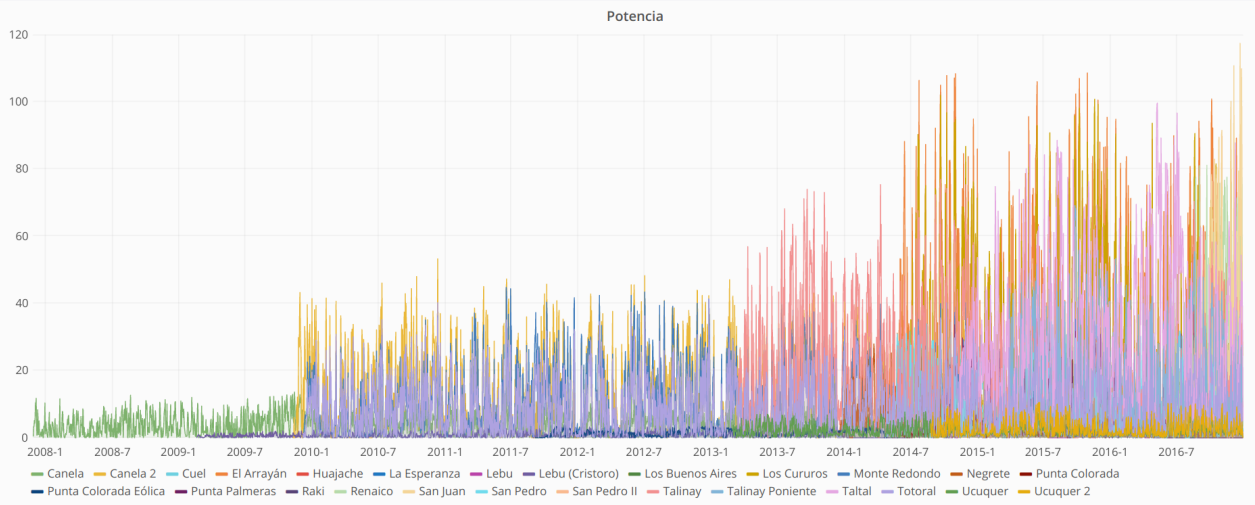

Figura 5.1: Potencial Real de las plantas eólicas informada por el CDEC

De todas las plantas eólicas se seleccionó sólo las con mayor tiempo de funcionamiento, esto debido a que para obtener buenos resultados con los modelos de RNN se necesita una gran cantidad de datos. Las plantas seleccionadas fueron Canela, Monte Redondo y Totoral. Todas estas plantas

se encuentran situadas en la cuarta región a distancias considerables para que existan variaciones climáticas entre las distintas plantas como se observa en la Figura 5.2.

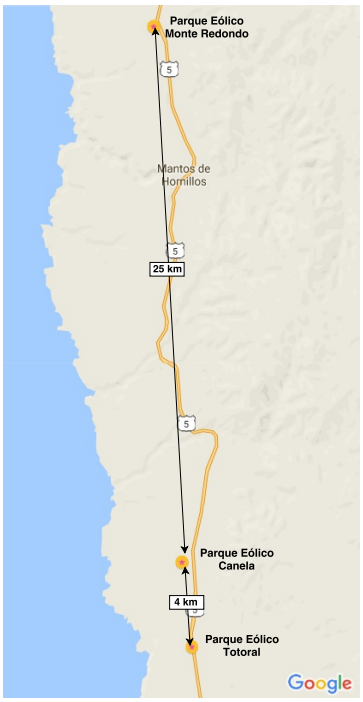

Figura 5.2: Mapa de localización de plantas seleccionadas

Canela es la primera planta eólica Chilena, su operación comenzó a inicios del año 2008, en el caso de Totoral y Monte Redondo ambas comenzaron su operación el año 2010, por lo que se tienen 9 y 7 años de datos respectivamente. En total se tienen entre 62 y 67 mil puntos de medición horaria.

En cuanto a la capacidad de cada planta, Canela cuenta con 11 Turbinas de 1.65 MW, dando una Potencia instalada de 18.15MW. Totoral tiene 23 Turbinas de 2MW, con una Potencia instalada de 46MW y Monte Redondo por su parte cuenta con 24 Turbinas de 2 MW, y una Potencia instalada de 48MW.

6 | Experimentación y Resultados.

Para realizar un análisis comparativo entre los distintos modelos expuestos en el capitulo 3 y 4 se comenzó con un análisis exploratorio de los datos, esencial para los modelos del tipo ARIMA y SARIMA debido a los supuestos utilizados para realizar el ajuste y posteriormente la predicción de los valores futuros, como también para la selección de algunos parámetros estructurales de dichos modelos, además nos permite tener una visión general de cómo es el comportamiento de este tipo de series.

Al momento de realizar la comparación se utilizó como punto de referencia, el **modelo naive**, donde el valor de predicción es igual al último valor observado

$$y^{t+h} = y^{(t)}, \tag{6.1}$$

y para cada modelo se calculó el RMSE (ecuación 4.22) en cada horizonte de tiempo $t + h$ con $h = 1 \ldots 12$, y se utilizó la media $\overline{RMSE_{t+h}}$ como métrica global del modelo.

Para el análisis exploratorio y posteriormente los experimentos se utilizo el 80 % de los datos para ajustar los modelos y 20 % como datos de prueba, suponiendo que el comportamiento de la serie es similar en ambos conjuntos. La división de los datos fue pensada para tener una gran cantidad de datos para el ajuste, dejando por lo menos un año de datos para validar los modelos y su poder de generalización.

Con respecto a la implementación de los modelos a comparar se utilizaron las implementaciones de ARIMA y SARIMA en la biblioteca *forecast* de R, en el caso de los modelos RNN se utilizaron la implementaciones en *Python* con la biblioteca *Keras* y *Tensorflow* para los modelos con LSTM y una implementación propia de las *ESN*.

Estos experimentos fueron ejecutados en el Cluster del Centro Científico Tecnológico de Valparaiso (CCTVAL) y en los equipos del laboratorio de Inteligencia Computacional Aplicada (INCA) de la UTFSM que cuentan con dispositivos GPU para acelerar el entrenamiento de los modelos con LSTM.

6.1. Análisis Exploratorio

Al observar los estadísticos globales (medias, varianzas, cuantiles, etc.) en la tabla 6.1 y el boxplot (Figura 6.1), se observa que los conjuntos de datos tienen escalas distintas, pero están relacionadas a la potencia instalada en cada planta.

	Canela	Monte Redondo	Totoral
n-ajuste	50083	49545	49545
n-prueba	12521	12387	12387
μ	3.02	11.44	9.23
σ	3.84	12.62	10.93
min	0.00	0.00	0.00
25 %	0.00	0.40	0.10
50 %	1.40	6.40	4.90
75 %	4.80	19.70	14.80
max	19.00	47.00	44.30

Tabla 6.1: Tabla resumen de los distintos Datasets utilizados

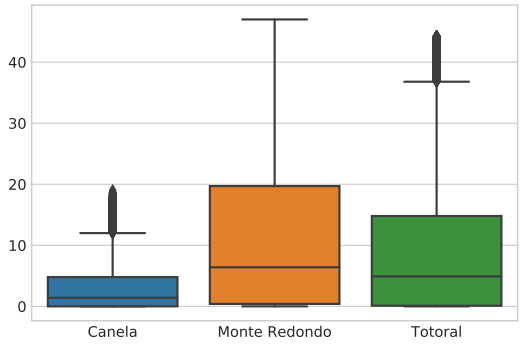

Figura 6.1: Boxplot de los distintos Datasets utilizados

Los estadísticos globales no nos otorgan información sobre la estacionariedad de la serie, por lo que se analizó el comportamiento de la media y varianza de la serie a través del tiempo (Figura 6.2). A simple vista se observa que la serie no tiene un comportamiento estacionario pues la media y varianza varían a través del tiempo, por lo que se intuye que la distribución finito dimensional del proceso subyacente varia.

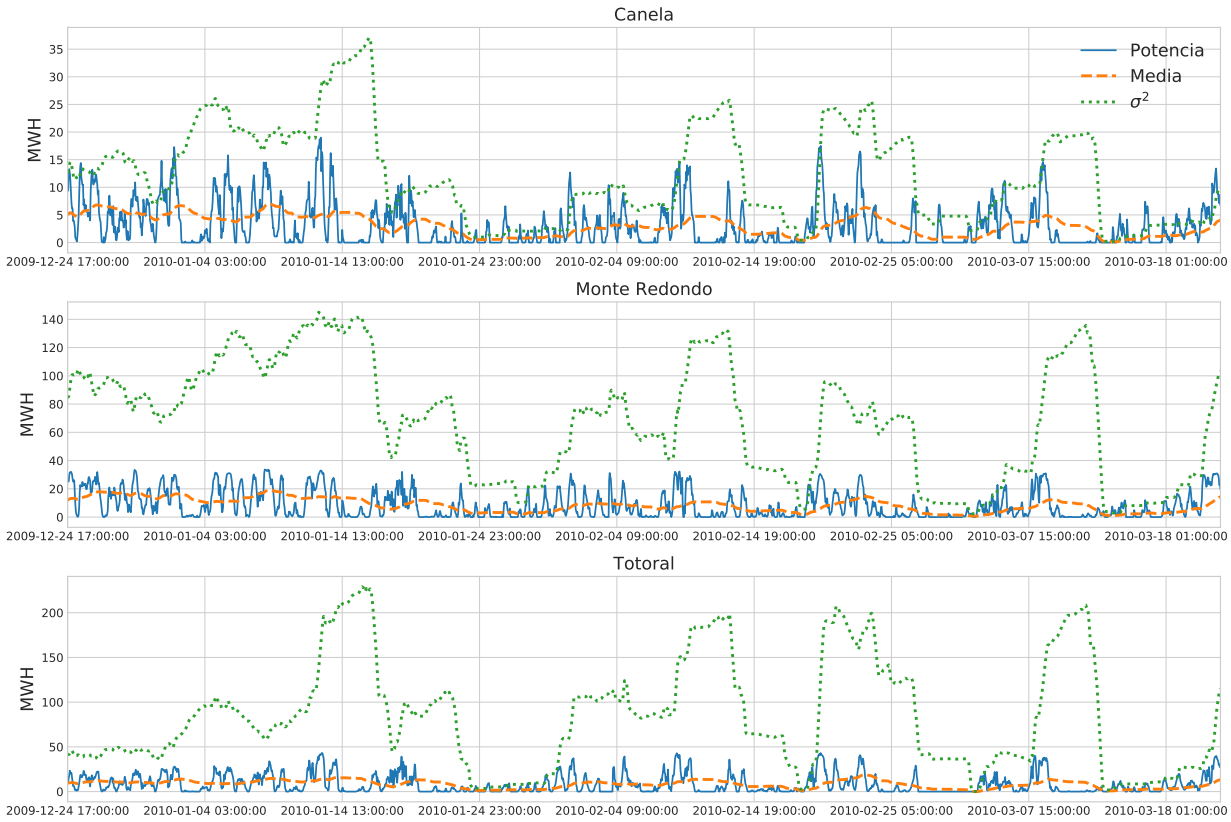

Figura 6.2: Media y varianza mediante una ventana deslizante de tamaño $n = 100$ a través del tiempo en los distintos conjuntos de datos

Al observar el diagrama de dispersión entre los lags x_t y x_{t-h} (Figura 6.3), se observa que existe una dependencia temporal lineal entre instantes cercanos. Esta dependencia lineal podría llevar a utilizar un modelo $AR(p)$ pero se violaría el supuesto de proceso estacionario.

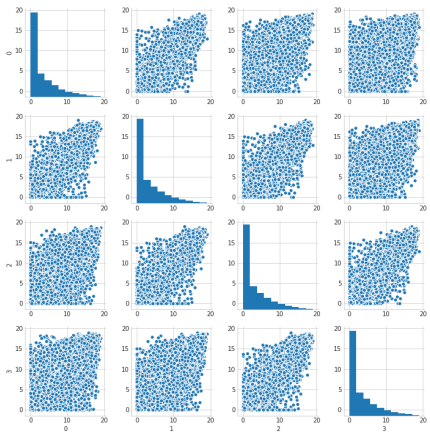

Figura 6.3: Diagrama de dispersión de Canela comparando $t - i$ vs $t - j$, en la diagonal ($i = j$) se encuentra el histograma de los datos.

Como se detalló en la sección 3.3.1, para determinar si la serie puede ser modelada como un proceso $AR(p)$, se puede analizar el gráfico de ACF y PACF. Al analizar el gráfico de ACF y PACF (Figura 6.4), se observa que el ACF disminuye lentamente y la gráfica de PACF luego del primer lag tiene valores cercanos a cero, por lo que el proceso puede ser un $AR(p)$, pero además se observa un comportamiento cíclico cada $S = 12$ lags en ambas gráficas, por lo que se intuye que la serie tiene una estacionalidad que puede ser modelada con un $SARIMA$.

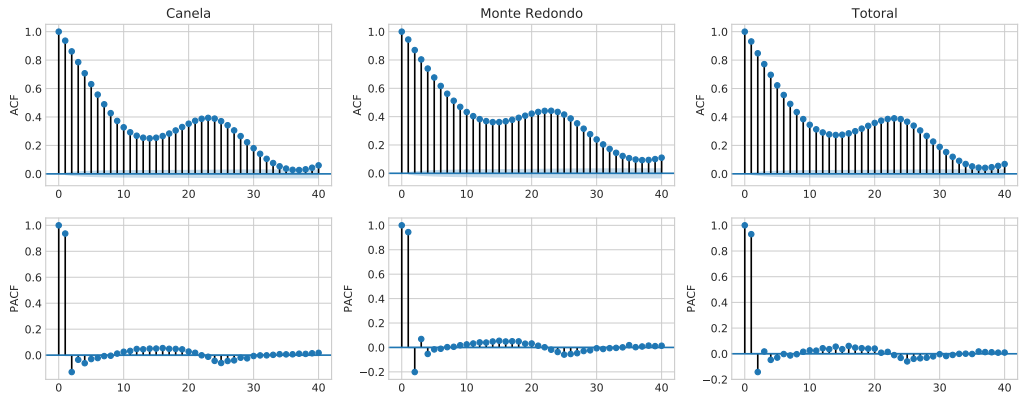

Figura 6.4: Gráficos de Auto-correlación y Auto-Correlación Parcial, en el eje x se encuentran los lags o instantes a futuro o pasado y en el eje y la correlación

Una posible transformación que ayuda a la estacionariedad de la serie es la ocupada por los modelos $ARIMA$, que realizan la diferenciación de la serie. Al aplicar la diferenciación con $d = 1$ (Figura 6.5), se observa que la media de la serie a través del tiempo se estabiliza cercana a cero, no es el caso de la varianza que todavía varía a través del tiempo.

Luego de realizar el análisis exploratorio no se observan características definitivas para definir los parámetros estructurales del modelo, pero se puede asumir que será un modelo $SARIMA$ con $S = 12$, donde el parámetro d de la componente no estacional del modelo es igual a 1.

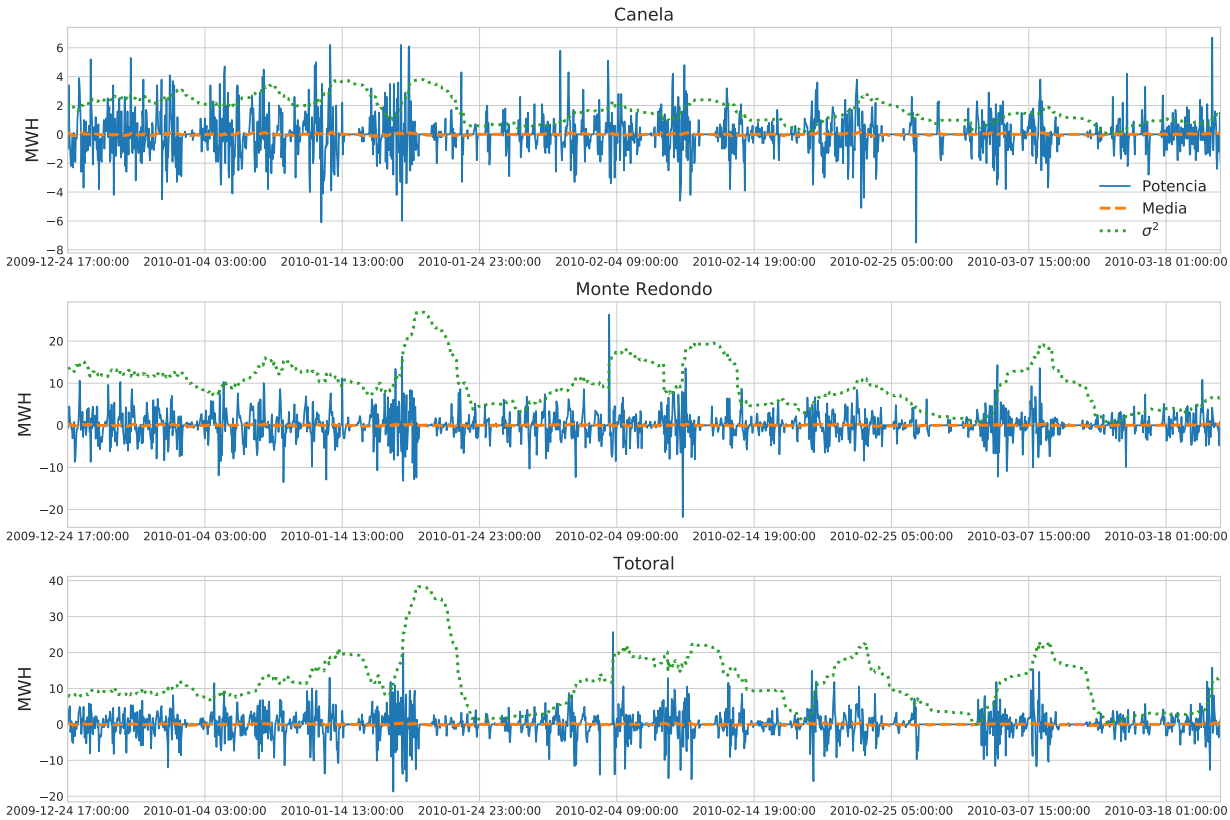

Figura 6.5: Media y varianza mediante una ventana deslizante de tamaño $n = 100$ a través del tiempo de datos diferenciados

6.2. Modelos ARIMA y SARIMA

Al realizar el análisis de datos se pudo caracterizar la serie, por su comportamiento se intuye que el proceso estocástico subyacente es estacional con $S = 12$ y tiene que ser modelado con $d = 1$, pero no se pueden determinar todos los parámetros estructurales para los modelos $ARIMA$ y $SARIMA$.

Para seleccionar los parámetros estructurales se utilizó el método alternativo de la sección 3.3.1, que utiliza el criterio AIC (ecuación 3.17), la estrategia utilizada para definir los parámetros estructurales se define a continuación.

1. Para ambos modelos se fijó $d = 1$. En el caso del modelo $SARIMA$ además se fijó $S = 12$.

2. Se ajustaron modelos variando sus parámetros estructurales según:

   - $p + q \leq 4$ en el caso del modelo $ARIMA$.
   - $p + q \leq 4$, $P + Q \leq 4$ y $D \in \{0, 1\}$ en el modelo $SARIMA$.

3. En cada modelo se calculó su AIC con los datos de ajuste.

4. Se seleccionó el modelo $ARIMA$ con menor AIC, por su parte de todos los modelos $SARIMA$ se tomó el 5 % de los modelos con menor AIC, y se seleccionó el modelo con menor grado $p + q + P + Q$.

el paso 4 de los modelos $SARIMA$ se utiliza debido a que mientras más complejo el modelo, es decir mayor cantidad de parámetros tiene, mayor va a ser su tiempo de ajuste.

De esta forma se puede seleccionar el modelo que mejor ajuste a los datos, pero utilizando el modelo menos complejo (en términos de la cantidad de parámetros del modelo).

6.2.1. Resultados ARIMA y SARIMA

Antes de seleccionar los parámetros estructurales de cada modelo, se realizó análisis para validar la estrategia utilizada para la selección de los parámetros, en especial el uso de AIC y limitar el orden de los modelos.

Utilizando los resultados de los modelos $ARIMA$ que se ajustaron, podemos observar que el uso de AIC para la selección del modelo es adecuado ya que mientras menor AIC, menor es el MSE que tendrá el modelo (Figura 6.6), métrica que se utilizará para la comparación entre los modelos. Además se observa que mientras mayor es el grado del modelo, se ajustan mejor los datos, pero el tiempo de ajuste aumenta exponencialmente (Figura 6.7), por lo que seleccionar el modelo con menor orden además reduce su costo computacional.

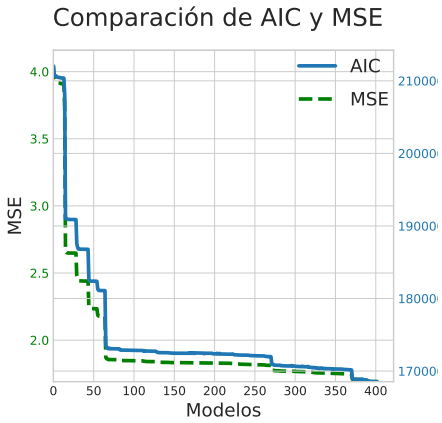

Figura 6.6: Comparación entre AIC y Error Cuadrático Medio en Canela

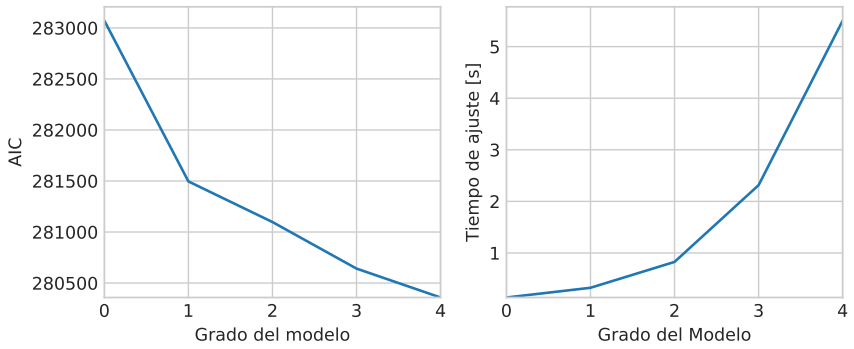

Figura 6.7: AIC y tiempo de Ajuste dependiendo del Grado del modelo tipo ARIMA

Luego de analizar los resultados generales, se seleccionó el mejor modelo $ARIMA$ y $SARIMA$ para cada conjunto de datos, los modelos seleccionados son los siguientes:

- Canela:
  - $ARIMA(4, 1, 0)$.
  - $SARIMA(1, 1, 2) \times (1, 0, 1)_{12}$.
- Monte Redondo:
  - $ARIMA(2, 1, 2)$.
  - $SARIMA(1, 1, 3) \times (1, 0, 1)_{12}$.
- Totoral:
  - $ARIMA(2, 1, 2)$.
  - $SARIMA(3, 1, 1) \times (1, 1, 2)_{12}$.

Se observa que el modelo $ARIMA$ de Canela es un modelo $ARI(p, d)$, que se esperaba luego de realizar el análisis exploratorio, en cambio los modelos $ARIMA$ de Monte Redondo y Totoral no son $AR(p)$, sino que son modelos $ARIMA(p, d, q)$. Esto se puede deber a que las series no son estacionarias. Por otro lado al analizar los modelos $SARIMA$ que se seleccionaron, se observa que Canela y Monte Redondo tienen el mismo componente estacional,cuya causa puede ser que el comportamiento estacional se debe al comportamiento cíclico de la velocidad del viento y no a la operación de la planta, aunque existe la posibilidad de que sea simplemente una coincidencia dado que Canela se encuentra alejado de Monte Redondo, y su planta más cercana (Totoral) no tiene el mismo comportamiento cíclico.

Al analizar el gráfico de $RMSE$ en los distintos horizontes de tiempo (Figura 6.8) se observa que la predicción a un paso de ambos modelos es similar a un modelo *Naive*, pero a medida que aumenta el horizonte de predicción los modelos $SARIMA$ obtienen mejores resultados. En el caso del modelo $ARIMA$ de Canela se observa que tiene resultados casi idénticos a un modelo *Naive*, por lo que la predicción realizada por el modelo no va a ser útil.

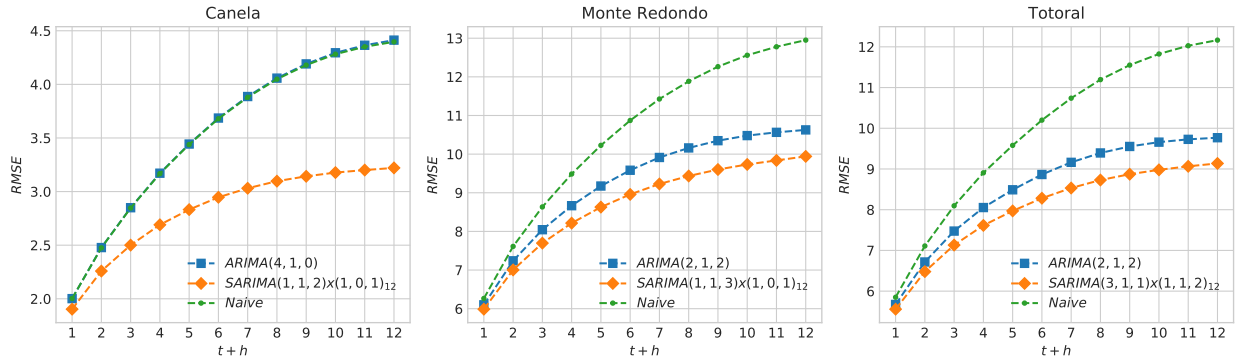

Figura 6.8: RMSE de los modelos $ARIMA$ y $SARIMA$

El análisis de la media de $RMSE$ (tabla 6.2) entrega la misma información que el gráfico en los distintos horizontes (Figura 6.8), se observa que la diferencia entre el modelo $ARIMA$ y *Naive* en Canela es despreciable, por otro lado en todos los conjuntos el modelo $SARIMA$ obtiene mejores resultados, debido a la estacionalidad de la serie.

	Canela	**Monte Redondo**	**Totoral**
Naive	3.56	10.58	9.94
$ARIMA$	3.57	9.24	8.54
$SARIMA$	**2.83**	**8.69**	**8.03**

Tabla 6.2: Media de RMSE de los modelos $ARIMA$ y $SARIMA$

6.3. Modelos RNN

Para la selección de los modelos de RNN se utilizó la metodología propuesta en la sección 4.3, por lo que se dividió el conjunto de entrenamiento en 80 % para ajustar el modelo y 20 % para validación. Con el nuevo conjunto de entrenamiento se ajustaron los modelos variando el preprocesamiento y los hiper-parámetros del modelo, como ambos modelos tienen distintos hiper-parámetros se detallaran en su sección correspondiente (secciones 6.3.1 y 6.3.2).

Como los conjuntos de datos utilizados no tienen datos faltantes, el preprocesamiento sólo fue la transformación de la serie. Las transformaciones utilizadas fueron estandarización y normalización en los rangos $[0, 1]$ y $[-1, 1]$.

Se realizaron experimentos variando el tamaño del vector de entrada o lags, utilizando una ventana deslizante de tamaño $I \in \{1, 6, 12, 24\}$, y cómo el modelo realiza la predicción, definido como **predicción múltiple** y **predicción recursiva**.

La predicción múltiple es donde el vector de salida de la RNN, es de la forma $y^{(t)} = \{y_1^{(t)}, y_2^{(t)}, \ldots y_h^{(t)}\}$, donde $y_i^{(t)} = y^{(t+i)}$.

Por su parte la predicción recursiva es similar a la predicción realizada por el modelo $ARIMA$ y $SARIMA$, el modelo RNN tiene un único valor de salida y recursivamente se utilizan las predicciones dentro del vector de entrada hasta un horizonte $t + h$.

6.3.1. Experimentos y resultados LSTM

Los hiper-parámetros relevantes dentro de la LSTM son la cantidad de *memory cells* en la capa oculta y las funciones de activación de los nodos y compuertas. Para simplificar la búsqueda del modelo se fijó la cantidad de *memory cells* en $n = 32$, y las funciones de activación que se utilizaron fueron:

- Nodos (g_c, h_c): Tangente Hiperbólica (tanh), Sigmoidal σ, y $ReLU$.
- Compuertas (i_c, f_c, o_c): Sigmoidal σ y una variación denominada *Hard Sigmoid*, que quita las suaves transiciones que realiza la funcion sigmoidal.

Otro parámetro que se analizará en los experimentos del modelo LSTM está relacionado al BPTT que se utiliza para ajustar los pesos de la red, este parámetro se denomina *timesteps*, y es la cantidad de instantes de tiempo que el gradiente se propaga en el tiempo, también se puede ver como cuantos instantes se desarrolla la recursión del modelo al momento de realizar el entrenamiento.

Los resultados globales obtenidos para el modelo LSTM con salida múltiple (tabla 6.3), nos muestran que el modelo LSTM obtiene mejores resultados utilizando una ventana deslizante de un valor ($I = 1$) o un lag, y mientras más se desarrolla la recursión menor es el RMSE, esto se puede deber a que la LSTM aprende dependencias a más largo plazo que le ayudan a realizar mejores predicciones.

		Canela				Monte Redondo				Totoral			
		Lags											
		1	6	12	24	1	6	12	24	1	6	12	24
Timesteps	1	2.83	2.81	2.74	2.66	8.92	8.88	8.65	8.37	8.09	8.07	7.91	7.77
	24	2.67	2.68	2.66	2.69	8.40	8.44	8.58	8.34	7.81	7.77	7.85	7.86
	72	2.64	2.63	2.67	2.69	**8.20**	8.34	8.31	8.43	7.75	7.79	7.82	7.75
	168	**2.55**	2.59	2.58	2.65	8.26	8.32	8.33	**8.20**	**7.54**	7.62	7.71	$5.69e^{13}$

Tabla 6.3: Media de RMSE de modelo LSTM con predicción múltiple, variando la cantidad de lags ($t - k$) de entrada y la cantidad de instantes (*timesteps*) que se desarrolla la RNN en el BPTT.

En el caso de Totoral, el RMSE obtenido con 24 lags y 168 timesteps, se puede deber a que el modelo seleccionado utiliza la función de activación *ReLU* para los nodos g_c y h_c, una función de activación que no está en el estado del arte del modelo LSTM, por lo que no se asegura que se evite el problema de *exploding gradient*.

Como se mostró que el modelo LSTM tiene buenos resultados con un lag (ventana deslizante de tamaño 1), para el modelo recursivo sólo se utilizó $I = 1$.

Al analizar los resultados del modelo LSTM recursivo (tabla 6.4), análogo al modelo múltiple se observa que a mayor cantidad de *timesteps*, se obtienen mejores resultados. Al analizar el gráfico de RMSE con respecto al horizonte de predicción (Figura 6.9), se observa que desde 72 *timesteps* se tiene la información necesaria a largo plazo para obtener buenos resultados.

Timesteps	Canela	Monte Redondo	Totoral
1	9.61	28.73	25.68
24	3.30	12.89	8.32
72	**2.66**	8.81	7.82
168	2.75	**8.29**	**7.70**

Tabla 6.4: Media de RMSE de modelo LSTM con predicción recursiva, utilizando 1 lag de entrada y variando la cantidad de instantes (*timesteps*) que se desarrolla la RNN en el BPTT.

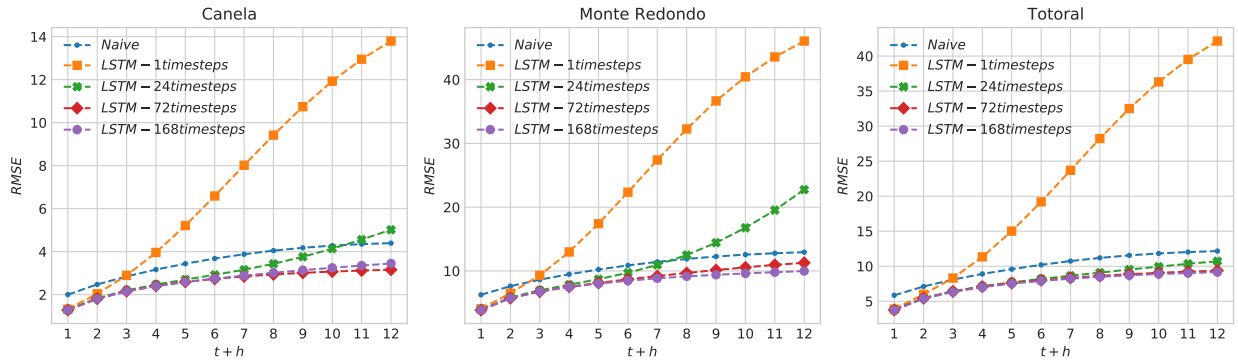

Figura 6.9: RMSE de cada horizonte de tiempo para el modelo LSTM con predicción recursiva.

Los resultados de ambos modelos son comparables, en general son levemente mejores en el modelo múltiple, pero esto limita el horizonte de predicción del modelo.

Finalmente el modelo seleccionado fue el de predicción múltiple con un lag y 168 *timesteps* para todos los conjuntos. Los hiper-parámetros de cada modelo se detallan en la tabla 6.5.

		Canela	**Monte Redondo**	**Totoral**
	Preprocesamiento	Estandarización	Estandarización	Estandarización
	memory cells	32	32	32
Función de activación	Nodos	tanh	tanh	tanh
	Compuertas	σ	σ	σ

Tabla 6.5: Hiper-parámetros seleccionados para modelo LSTM de predicción múltiple

6.3.2. Experimentos y resultados ESN

Siguiendo lo propuesto en [24], para la selección del modelo ESN se comenzó fijando el tamaño del reservorio en $N = 100$ para el proceso del selección de hiper-parámetros, luego al momento de entrenar el modelo seleccionado se fijó el tamaño del reservorio en $N = 1000$. Los hiper-parámetros que se variaron fueron:

- *sparsity* $\in$ [0.5,0.9].
- *leaking rate* $a \in$ [0.3,0.9].
- Regularización (Ridge) $\alpha \in [10^{-4}, 10^{4}]$.
- Radio Espectral $\rho(W) \in [10^{-5}, 1]$.
- Función de activación: Tangente Hiperbólica (tanh), Sigmoidal (σ), $ReLU$.

A diferencia del modelo LSTM, la ESN cuenta con una gran cantidad de parámetros, por lo que la mayor cantidad del tiempo que se utiliza para este modelo es dedicada a la selección del modelo, más que al entrenamiento del modelo seleccionado.

Análogo al modelo LSTM, la ESN obtiene mejores resultados cuando se realiza una predicción múltiple, por otro lado la ESN en algunos casos se beneficia al utilizar una ventana deslizante de mayor tamaño, es el caso del modelo seleccionado en Monte Redondo que el mejor resultado se obtiene con 24 lags (tabla 6.6).

	Canela		Monte Redondo		Totoral	
Modelo	Multi	Recursivo	Multi	Recursivo	Multi	Recursivo
Naive	3.56	3.56	10.58	10.58	9.94	9.94
ESN 1 lag	**2.63**	**3.01**	8.40	9.29	**7.52**	**8.36**
ESN 6 lags	2.63	3.10	8.21	9.32	7.86	8.65
ESN 12 lags	2.63	4.39	8.40	9.29	7.67	10.32
ESN 24 lags	2.69	3.07	**8.17**	**9.22**	7.61	8.58

Tabla 6.6: Media de RMSE del modelo ESN con predicción múltiple y recursivo, variando la cantidad de lags de entrada.

Si observamos los resultados del modelo con predicción recursiva, se observa que hay dos casos (Canela y Totoral con 12 lags) donde el RMSE es mayor al modelo Naive. Analizando el RMSE en cada horizonte (Figura 6.10), estos modelos obtienen buenos resultados en $t + 1$, pero a medida que

aumenta el horizonte la predicción empeora rápidamente, esto se puede deber a que el reservorio no llegó al estado echo, por lo que el modelo no tiene unicidad entre el estado y la predicción, haciendo oscilar la predicción.

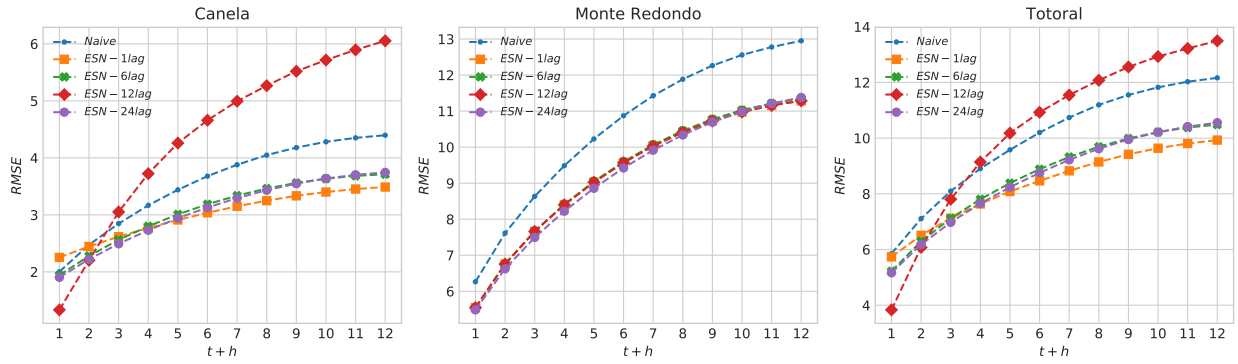

Figura 6.10: RMSE según el horizonte de predicción $t + h$ para el modelo ESN con predicción recursiva.

Luego de realizar los diversos análisis el modelo seleccionado en todos los casos fue el de predicción múltiple, con un lag para Canela y Totoral, y 24 lags para Monte Redondo. Los hiper-parámetros de cada modelo se detallan en la tabla 6.7.

	Canela	**Monte Redondo**	**Totoral**
Preprocesamiento	Normalización [0, 1]	Estandarización	Normalización [0, 1]
Tamaño Reservorio (N)	1000	1000	1000
sparcity	0.2	0.2	0.2
a	0.3	0.3	0.3
α	0.15	251	0.15
$\rho(W)$	1	1	1
Función de Activación	tanh	relu	tanh

Tabla 6.7: Hiper-parámetros seleccionados modelo ESN de predicción múltiple

6.4. Comparación de Modelos

Luego de realizar la selección de los modelos podemos realizar el análisis comparativo.

El RMSE a distintos horizontes de tiempo (Figura 6.11) muestra que los modelos de RNN tienen mejores resultados, en especial cuando el horizonte de predicción es corto (hasta $t+5$), esto se puede deber a que la serie tiene dependencias no lineales, que no pueden ser modeladas con $ARIMA$ y $SARIMA$ debido a ser modelos lineales.

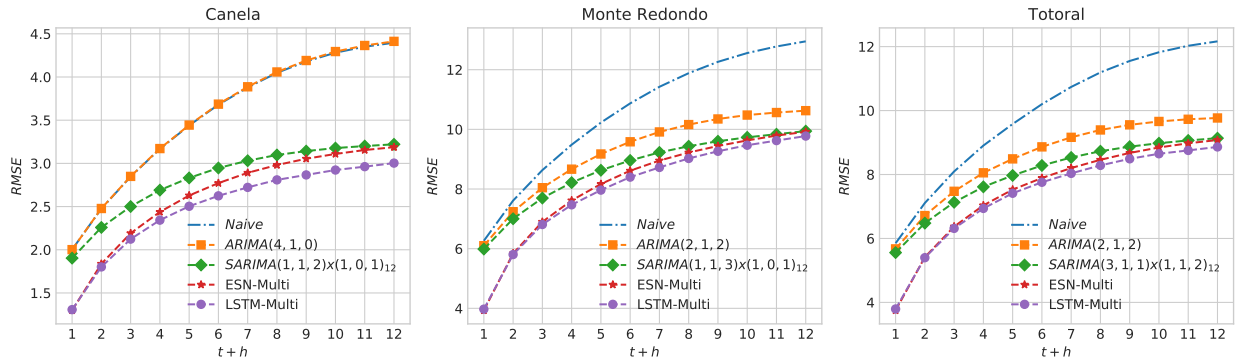

Figura 6.11: RMSE según el horizonte de predicción para los modelos seleccionados.

Para asegurar que la diferencia entre los RMSE del modelo $SARIMA$ y de los modelos RNN sea estadísticamente significativa, se utilizó un test de hipótesis, donde la hipótesis nula es que los MSE entre el modelo $SARIMA$ y de RNN son iguales.

Al analizar los $p - values$ resultantes con una confianza del 99 % (Figura 6.8), se puede asegurar que los modelos de RNN obtienen resultados estadísticamente significativos, hasta $t + 7$ en el caso de la ESN y $t + 10$ para el modelo RNN.

		$t+1$	$t+2$	$t+3$	$t+4$	$t+5$	$t+6$	$t+7$	$t+8$	$t+9$	$t+10$	$t+11$	$t+12$
Canela	LSTM	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**
	ESN	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.01**	0.04	0.11	0.28	0.44
Monte Redondo	LSTM	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.01**	0.04	0.11
	ESN	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.01**	0.04	0.13	0.32	0.66	0.83
Totoral	LSTM	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.01**
	ESN	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.0**	**0.01**	0.04	0.18	0.37	0.56

Tabla 6.8: $p - values$ Test de Hipótesis de diferencia estadística entre MSE de modelos SARIMA y modelos RNN seleccionados, se dice que existe una diferencia estadística si $p - value \leq \alpha$, se utilizó $\alpha = 0{,}01$

7 Conclusiones y trabajo futuro

El crecimiento acelerado que tienen las ERNC en la generación de energía en Chile, entrega un gran potencial para realizar investigación aplicada, y así facilitar el ingreso de estas energías al Sistema Eléctrico Nacional (SEN), reduciendo los costos del ingreso de las ERNC al SEN, que se reflejará en energía más económica para los chilenos. En especial la investigación en modelos de predicción de potencia generada, ya sea de forma directa o indirecta es un tema importante para ayudar en la toma decisiones informadas sobre cómo se administrará la potencia generada.

La predicción de la potencia eólica es un tema atractivo para investigar, como la serie de la velocidad del viento e indirectamente la serie de la potencia generada, tiene características que dificultan realizar un modelo estadístico de la serie, como la no linealidad y estacionalidad presentes en la serie hacen necesario recurrir a modelos estadísticos más complejos que tomen en cuenta estas características.

Luego de realizar experimentos para distintas plantas eólicas se puede afirmar que en el problema específico de predicción a 12 horas en la serie horaria de la potencia eólica generada, los modelos $SARIMA$ y de Redes Neuronales Recurrentes (RNN) tienen mejores resultados con respecto a la métrica RMSE.

El modelo $SARIMA$ al tomar en cuenta la componente estacional de la serie alcanza mejores resultados que el modelo *Naive* y $ARIMA$, además se mostró que dos conjuntos de datos tienen el mismo componente estacional, igualdad que se puede asociar a que gran parte del comportamiento cíclico es debido a la velocidad del viento más que a factores inherentes a la operación de la planta.

Por su parte los modelos de la familia RNN, obtienen mejores resultados que el modelo $SARIMA$. Esto es debido a que al momento de modelar la serie, los modelos RNN no realizan supuestos sobre la serie y además realizan un modelo altamente no lineal, por lo que pueden modelar series no lineales que no presentan estacionariedad, como es el caso de la serie de potencia eólica.

Entre los modelos de RNN, se mostró que la predicción múltiple obtiene mejores resultados a los distintos horizontes de predicción en comparación a la predicción recursiva. Esto se debe a que los nodos dentro de la red se especializan en realizar la predicción de un instante en específico, a diferencia del modelo recursivo, donde el error de la predicción se lleva a la predicción siguiente. Una desventaja del modelo de predicción múltiple es que se limita el horizonte de predicción del modelo, en cambio el modelo de predicción recursivo puede realizar una predicción a un horizonte arbitrariamente lejano pero perdiendo eficacia a medida que aumenta el horizonte.

Además se observó que la predicción del modelo RNN va a depender en gran medida de la arquitectura de la red. Se mostró que el modelo LSTM al poder aprender dependencias a más largo plazo, tiene mejores resultados a horizontes de predicción más largo; En comparación al modelo ESN que realizan predicciones comparables al modelo $SARIMA$ cuando el horizonte de predicción es muy largo.

7.1. Trabajo Futuro

Luego de realizar este trabajo nacieron varias interrogantes y posibles variaciones a los experimentos realizados que pueden ser útiles para mejorar los resultados.

Para comenzar es necesario realizar una investigación más a fondo sobre de otras variables que pueden afectar la generación eólica, como por ejemplo la temperatura, velocidad del viento, densidad del aire.

Por otro lado, estudiar si existe una correlación entre la potencia generada por una planta eólica con las plantas cercanas y utilizar *transfer learning* entregando información de las plantas cercanas al modelo.

Otro punto relevante es poder predecir la incertidumbre de la predicción del modelo, es decir realizar una predicción con intervalos de confianza con los modelos de la familia RNN, ya que sólo la media de la potencia generada es insuficiente al momento de tomar decisiones, por lo que es necesario tener una medida de la incertidumbre para poder tener un plan de contingencia si la potencia generada no es acorde a la predicción.

Bibliografía

[1] Braden Heffernan James Jenden Ellen Lloyd Jason Donev Allison Campbell, Jordan Hanania. Wind power, 2015. (document), 2.1

[2] Yoshua Bengio, Patrice Simard, and Paolo Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994. 4.1

[3] Fatih Birol et al. World energy outlook. *Paris: International Energy Agency*, 23(4):329, 2008. 1

[4] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015. 2.2

[5] Erasmo Cadenas, Wilfrido Rivera, Rafael Campos-Amezcua, and Christopher Heard. Wind speed prediction using a univariate arima model and a multivariate narx model. *Energies*, 9(2):109, 2016. 2.2

[6] CDEC. *Norma Técnica de Seguridad y Calidad de Servicio*. Gobierno de Chile, 2016. 1

[7] Peiyuan Chen, Troels Pedersen, Birgitte Bak-Jensen, and Zhe Chen. Arima-based time series model of stochastic wind power generation. *IEEE Transactions on Power Systems*, 25(2):667–676, 2010. 2.2

[8] Generadoras Chile. Reporte anual. 2016. (document), 1.1

[9] Comité Consultivo de Energía. 2050. *Ministerio de Energía,"Hoja de Ruta"*, 2050, 2014. 1

[10] Ali Deihimi and Hemen Showkati. Application of echo state networks in short-term electric load forecasting. *Energy*, 39(1):327–340, 2012. 2.2

[11] Fathalla A Eldali, Timothy M Hansen, Siddharth Suryanarayanan, and Edwin KP Chong. Employing arima models to improve wind power forecasts: A case study in ercot. In *North American Power Symposium (NAPS), 2016*, pages 1–6. IEEE, 2016. 2.2

[12] Jeffrey L Elman. Finding structure in time. *Cognitive science*, 14(2):179–211, 1990. 4.1

[13] A. A. Ferreira and T. B. Ludermir. Evolutionary strategy for simultaneous optimization of parameters, topology and reservoir weights in echo state networks. In *The 2010 International Joint Conference on Neural Networks (IJCNN)*, pages 1–7. IEEE, 2010. 2.2

[14] Felix A Gers and Jürgen Schmidhuber. Recurrent nets that time and count. In *Neural Networks, 2000. IJCNN 2000, Proceedings of the IEEE-INNS-ENNS International Joint Conference on*, volume 3, pages 189–194. IEEE, 2000. 4

[15] Ernesta Grigonyte and Egle Butkeviciute. Short-term wind speed forecasting using arima model. *Energetika*, 62(1-2), 2016. 2.2

[16] Kira Grogg. Harvesting the wind: the physics of wind turbines. 2005. 1

[17] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997. 4.1.1

[18] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982. 4.1

[19] Herbert Jaeger. Echo state network. *Scholarpedia*, 2(9):2330, 2007. 4.1.2

[20] Jaesung Jung and Robert P Broadwater. Current status and future advances for wind speed and power forecasting. *Renewable and Sustainable Energy Reviews*, 31:762–777, 2014. (document), 2, 2.2, 2.3

[21] Rajesh G Kavasseri and Krithika Seetharaman. Day-ahead wind speed forecasting using f-arima models. *Renewable Energy*, 34(5):1388–1393, 2009. 2.2

[22] Da Liu, Jilong Wang, and Hui Wang. Short-term wind speed forecasting based on spectral clustering and optimised echo state networks. *Renewable Energy*, 78:599–608, 2015. 2.2

[23] Erick López, Carlos Valle, and Héctor Allende. Recurrent networks for wind speed forecasting. 2016. 2.2

[24] Mantas Lukoševičius. A practical guide to applying echo state networks. In *Neural networks: tricks of the trade*, pages 659–686. Springer, 2012. 2.2, 6.3.2

[25] Michael Milligan, Marc Schwartz, and Yih-huei Wan. Statistical wind power forecasting models: Results for us wind farms; preprint. Technical report, National Renewable Energy Laboratory (NREL), Golden, CO., 2003. 2.2

[26] Niels Gylling Mortensen, Duncan N Heathfield, Lisbeth Myllerup, Lars Landberg, and Ole Rathmann. Wind atlas analysis and application program: Wasp 9 help facility, 2007. 2.1

[27] Mark Pinsky. Stochastic differential equations. an introduction with applications (b. oksendal), 1987. 3.1

[28] WindPower project. Wind turbine power ouput variation with steady wind speed., 2010. 2.1

[29] Virginijus Radziukynas and Arturas Klementavicius. Short-term wind speed forecasting with arima model. In *Power and Electrical Engineering of Riga Technical University (RTUCON), 2014 55th International Scientific Conference on*, pages 145–149. IEEE, 2014. 2.2

[30] Jing Shi, Xiuli Qu, and Songtao Zeng. Short-term wind power generation forecasting: Direct versus indirect arima-based approaches. *International Journal of Green Energy*, 8(1):100–112, 2011. 2.2

[31] William C Skamarock, Joseph B Klemp, Jimy Dudhia, David O Gill, Dale M Barker, Wei Wang, and Jordan G Powers. A description of the advanced research wrf version 2. Technical report, National Center For Atmospheric Research Boulder Co Mesoscale and Microscale Meteorology Div, 2005. 2.1

[32] Saurabh S Soman, Hamidreza Zareipour, Om Malik, and Paras Mandal. A review of wind power and wind speed forecasting methods with different time horizons. In *North American Power Symposium (NAPS), 2010*, pages 1–8. IEEE, 2010. 2.1

[33] Core Writing Team, Rajendra K Pachauri, and Andy Reisinger. Climate change 2007: synthesis report. *Geneva, Switzerland: IPCC*, 104, 2007. 1