



CENTRO DE INVESTIGACIÓN Y DE ESTUDIOS AVANZADOS
DEL INSTITUTO POLITÉCNICO NACIONAL

Unidad Zacatenco

Departamento de Computación

Predicción de energía eólica usando modelos híbridos múltiples de ARIMA

Tesis que presenta

Vicente Adnan Duarte Méndez

para obtener el Grado de

Maestro en Ciencias en Computación

Director de la Tesis

Dra. Xiaoou Li

Ciudad de México

Diciembre 2022

Resumen

La predicción de energía eólica presenta grandes desafíos como la integración de la red, la calidad de la energía, entre otros. Varios de estos problemas pueden arreglarse con una calendarización óptima de energía, lo que implica que se requiere pronósticos precisos y confiables, sin embargo, no todos los modelos de predicción otorgan estas cualidades. En la predicción de energía eólica existen modelos matemáticos estadísticos y físicos. Estos modelos proveen buena confiabilidad, pero la gran mayoría no puede trabajar con series de tiempo de pocos datos, debido a que, al utilizarlos, se crean modelos con poca precisión y pronósticos inciertos. En tesis propone construir un modelo múltiple de ARIMA (del inglés autoregressive integrated moving average) para mejorar la precisión de un solo modelo óptimo ARIMA de un conjunto presentado. El modelo múltiple se construye con diferentes series de tiempo de energía generada obtenida en granjas eólicas situadas en California y Alemania. Se calcula el modelo ARIMA óptimo para cada conjunto mediante el criterio AIC (Criterio de Aikake), se selecciona un conjunto de datos a evaluar y se obtienen los residuales ARIMA de todos los modelos. Estos fungirán como características a la entrada de una red neuronal para pre-entrenarla y encontrar un modelo completamente nuevo. El objetivo es obtener un modelo híbrido que mejore la precisión en el pronóstico en comparación a la red neuronal y el modelo ARIMA.

Abstract

Wind power forecasting presents great challenges such as grid integration, power quality, among others. Several of these problems can be fixed with optimal energy scheduling, which implies that accurate and reliable forecasts are required, however, not all forecast models provide these qualities. In the prediction of wind energy there are statistical and physical mathematical models. These models provide good reliability, but the vast majority cannot work with time series of few data, because, using them, models with little precision and uncertain forecasts are created. In thesis he proposes to build a multiple ARIMA (autoregressive integrated moving average) model to improve the accuracy of a single optimal ARIMA model from a presented set. The multiple model is built with different time series of generated energy obtained in wind farms located in California and Germany. The optimal ARIMA model for each set is calculated using the AIC criterion (Aikake's Criterion), a data set to be evaluated is selected and the ARIMA residuals of all the models are obtained. These will serve as features at the input of a neural network to pre-train it and find a completely new model. The objective is to obtain a hybrid model that improves forecast accuracy compared to the neural network and the ARIMA model.

Agradecimientos

A CONACYT por el apoyo económico que me brindo para estudiar la maestría, al CINVESTAV por el conocimiento y el lugar para poder desarrollar investigación de calidad.

A mis padres y hermanas que me han brindado su apoyo durante toda mi vida y me ayudaron en mi decisión de seguir por el camino de la investigación.

A mi tutora y asesora Dra. Xiaoou Li, por aguantarme y llevarme por el camino correcto para realizar mis investigaciones y esta tesis.

A los Doctores que me impartieron clases, sinodales y demás profesores, gracias por los consejos y la atención que me brindaron para aprender nuevos temas.

A mis amigos en general que nunca me dejaron de apoyar y creyeron en mí.

Índice general

Resumen	III
Abstract	V
Agradecimientos	VII
Índice de figuras	X
Índice de tablas	XII
1. Introducción	1
1.1. Enfoque de las energías renovables	1
1.2. Motivación y planteamiento del problema	4
1.3. Objetivo y contribución de la tesis	5
1.4. Organización de la tesis	6
2. Fundamentos	7
2.1. Energía eólica	7
2.2. Predicción del viento: descripción general	9
2.3. Series de tiempo	12
2.4. Análisis de series estacionarias	15
2.4.1. Comprobación de serie estacionaria (Prueba Dickey-Fuller Au- mentada)	15
2.4.2. Proceso estocástico	17
2.4.3. Funciones de auto-covarianza y auto-correlación	20
3. Modelos para predicción de series de tiempo	23
3.1. Modelo lineal para series de tiempo estacionarias	23
3.1.1. Procesos Autorregresivos AR	24
3.1.2. Procesos de medias móviles MA	25
3.1.3. Procesos mixtos autoregresivos-promedio móvil	26
3.2. Modelo lineal para series de tiempo no estacionarias	27
3.2.1. Método Autorregresivo integrado de media móvil	27
3.3. Construcción del modelo múltiple ARIMA	29
3.3.1. Diseño de la estructura del modelo ARIMA	30

3.3.2.	Identificación del coeficiente del modelo ARIMA	31
3.3.3.	ARIMA múltiple	32
4.	Modelo de aprendizaje automático para predicción	33
4.1.	Redes neuronales	33
4.1.1.	Perceptrón	35
4.1.2.	Red neuronal artificial multicapa	39
4.1.3.	Consideraciones a tomar en el diseño de una red neuronal	43
4.1.4.	Características de una red neuronal	43
4.2.	Métricas de estimación	44
5.	Modelo híbridos	47
5.1.	Modelos híbridos	47
5.2.	Modelo híbrido múltiple ARIMA	52
5.2.1.	ARIMA múltiple	52
5.2.2.	Transferencia de aprendizaje en múltiples modelos ARIMA	52
5.2.3.	ARIMA múltiple combinado con redes neuronales	53
6.	Pronóstico de energía eólica con modelo híbrido múltiple ARIMA	57
6.1.	Conjuntos de datos	57
6.2.	Pre-procesamiento de datos	61
6.3.	Prueba estacionaria	63
6.4.	Entrenamiento de los modelos ARIMA	65
6.5.	Predicción del modelo ARIMA múltiple	69
6.5.1.	Predicción del conjunto de prueba para granja California-ISO	70
6.5.2.	Predicción del conjunto de prueba para granja Tennet-Alemania	71
6.5.3.	Predicción del conjunto de prueba para granja 1 Kaggle	72
7.	Conclusiones y trabajos futuros	75
7.1.	Conclusiones	75
7.2.	Trabajo a futuro	77
	Bibliografía	77

Índice de figuras

2.1. Tipos de clasificación de predicción de energía eólica.	9
2.2. Ejemplo de una serie de tiempo, donde z_t es el rendimiento de 100 repeticiones de un proceso aleatorio.	12
2.3. Componentes de una serie de tiempo.	13
2.4. Series de tiempo de datos adquiridos de los parques eólicos de California.	14
2.5. Gráfica para Dickey-Fuller aumentado	17
2.6. Campo z_t de 70 repeticiones de un proceso aleatorio.	19
2.7. Diagramas de Scatter.	20
2.8. Matriz de autocorrelación y correspondencia de función de autocorrelación para proceso estacionario.	21
2.9. Función de autocorrelación para mitad positiva de la matriz.	21
3.1. Arquitectura del modelo ARIMA para pronóstico de series de tiempo.	32
4.1. Función booleana modelada mediante un perceptrón.	35
4.2. Frontera de decisión del perceptrón.	38
4.3. Problema de clasificación de una compuerta XOR.	38
4.4. Ejemplo de una red neuronal multicapa.	39
4.5. Funciones de activación para redes neuronales.	40
4.6. Problema XOR visto desde una red neuronal de dos capas.	41
4.7. Superficie de error $E(w_1, w_2)$ para el modelo de 2 parámetros.	42
5.1. Modelo propuesto por Cadenas et al. 2010	48
5.2. Modelo propuesto por Liu et al. 2012	48
5.3. Modelo propuesto por Yan-Fei et al. 2015	49
5.4. Modelo propuesto por Dimitru et al. 2019	51
5.5. Esquema de combinación del modelo ARIMA y ANN.	54
5.6. Esquema de combinación del modelo ARIMA y ANN con compensación.	55
6.1. Datos de Kaggle y visualización del formato de fechas.	58
6.2. Datos de California y visualización del formato de fechas.	59
6.3. Datos de Tennet-Alemania y visualización del formato de fechas.	60
6.4. Serie de tiempo normalizada de los datos California del periodo comprendido entre 1 de septiembre de 2011 hasta 31 de agosto del 2012.	62

6.5. Serie de tiempo normalizada de los datos Alemania-Tennet del periodo comprendido entre 23 de agosto de 2019 hasta 22 de septiembre del 2020.	62
6.6. Serie de tiempo normalizada de los datos Kaggle (Granja 1) del periodo comprendido entre 1 de septiembre de 2010 hasta 30 de agosto del 2011.	63
6.7. ACF/PCF Análisis para el conjunto de datos de Alemania.	66
6.8. ACF/PCF Análisis para el conjunto de datos de California.	66
6.9. ACF/PCF Análisis para el conjunto de datos de Kaggle.	67
6.10. Resultado de predicción de los modelos con los datos de California. .	70
6.11. Resultado de predicción de los modelos con los datos de Alemania. . .	71
6.12. Resultado de predicción de los modelos con los datos de Kaggle. . . .	73
6.13. Modelo de ARIMA-múltiple compensado	74

Índice de tablas

6.1. Resultado de la prueba Dickey-Fuller en los conjuntos de datos	63
6.2. Resumen de los parámetros ARIMA candidatos obtenidos con las pruebas de ACF, PAF y la prueba Dickey-Fuller aumentada.	67
6.3. AIC de los modelos ARIMA para los conjuntos de datos	68
6.4. Métricas para los modelos de California	70
6.5. Métricas para los modelos de Alemania	72
6.6. Métricas para los modelos de Kaggle	73
6.7. Métricas para los modelos Kaggle y el modelo compensado	74

Capítulo 1

Introducción

Hoy en día, las fuentes de energía alternativas han ganado mucha importancia debido a que garantizan un funcionamiento que no contamina el medio ambiente. Con recursos nucleares y combustibles fósiles limitados, las tecnologías de energía solar y eólica han superado su cuota de mercado. Además de crear un impacto ambiental positivo, la energía eólica ha abierto nuevas oportunidades de empleo a nivel mundial[1]. Desde la revolución industrial, la generación de energía para el aprovechamiento de las cadenas productivas y el consumo de la población humana es casi exclusivamente a partir de combustibles fósiles y derivados de ellos. Algunos de los problemas relevantes son el cambio climático, la contaminación del aire, la contaminación del suelo, los problemas de salud de la población cercana a las plantas generadoras[2]. El medio ambiente se ha visto afectado por los procesos productivos industriales humanos y por el uso indiscreto de los recursos energéticos. Es urgente cambiar el paradigma de la generación de energía a métodos alternativos para abandonar gradualmente el uso de combustibles fósiles y favorecer métodos más amigables con el medio ambiente.

1.1. Enfoque de las energías renovables

La evolución de las tecnologías de energía renovable durante la última década ha trascendido todas las expectativas. La capacidad instalada mundial y la producción de todas las tecnologías renovables han aumentado sustancialmente, las políticas de apoyo han seguido extendiéndose a más países en todas las regiones del mundo. A principios de la década de 1970 cuando la crisis energética y el colapso económico dieron como resultado el crecimiento de las fuentes de energía renovable, países en desarrollo como Dinamarca, Estados Unidos, España y Alemania iniciaron el negocio de las energías renovables debido al cambio climático.

Una fuente de energía renovable es aquella cuya viabilidad tangible sigue siendo ilimitada además de poder utilizarse en su forma bruta. Se considera “sostenible” si conserva los rasgos sociales, económicos y ambientales para las generaciones futuras.

Dado que las reservas de carbón son limitadas, las fuentes alternativas de energía han ingresado al mercado para proporcionar una ruta sostenible y económica para consumir energía limpia. La producción de energía a partir de fuentes renovables está aumentando a un ritmo constante y ha llevado al crecimiento económico en muchos países en desarrollo. Una de las principales razones para promover las fuentes de energía renovable es la preocupación ambiental que causa el agotamiento de la capa de ozono en la atmósfera y las emisiones de gases de efecto invernadero. Como resultado de las incesantes actividades industriales, surgió el primer tratado internacional, el Protocolo de Kioto, una extensión de la Convención Marco de las Naciones Unidas sobre el Cambio Climático (CMNUCC) con el lema de frenar las emisiones de gases artificiales [3]. Estas emisiones de gases incluyen dióxido de carbono (CO_2), óxido nitroso (N_2O), metano (CH_4) y hexafluoruro de azufre (SF_6). Además, las partes se han comprometido a reducir las emisiones a nivel nacional. Sin embargo, existen algunos países que no se comprometen con esta meta debido a su bajo ingreso per cápita. También existen algunas excepciones a las emisiones en forma de transporte marítimo internacional y aviación. La forma más limpia de producción de energía a partir de fuentes renovables sigue floreciendo en el sector energético, lo que genera enormes inversiones de los sectores público y privado.

El recuento de instalaciones provenientes de fuentes renovables se ha duplicado desde 2007. Entre las fuentes renovables, las energías eólica y solar han sido un foco de atención para los legisladores que creen que el viento es un combustible limpio. 2015 fue testigo de un aumento del 44 % en la energía eólica en comparación con 2014. Con la tecnología adecuada disponible, la energía de fuentes de energía renovables puede aprovecharse adecuadamente. Las tecnologías de energía hidroeléctrica, solar y eólica se consideraron los principales impulsores del mercado. Con la energía eólica marina cada vez más dominante en el mercado competitivo, se ha observado una disminución constante en el costo nivelado de la energía (LCOE). China representa la producción de energía más grande del mundo a partir de fuentes de energía renovable, una asombrosa capacidad instalada de 647 GW, que consiste en 313 GW de energía de fuentes hidroeléctricas. Países como Alemania, Suecia y Dinamarca se centran en integrar proporciones cada vez mayores de energía solar fotovoltaica y eólica en los sistemas de redes de servicios públicos mejorando las políticas reguladoras e incorporando un sistema de transmisión mejorado para ampliar las áreas de equilibrio. Mientras tanto, los países también están invirtiendo fuertemente en almacenamiento de energía para reducir un impacto significativo de la intermitencia [1]. Las fuentes de energía renovable han tenido un impacto significativo en las economías en desarrollo. Específicamente, la energía eólica ha estado en el centro de atención debido a su naturaleza limpia y ecológica. Disponible en forma libre, el régimen del viento para un terreno en particular depende esencialmente de variables climáticas como la temperatura, la presión y la humedad del aire ambiente. Sin embargo, la energía eólica padece de desafíos como la integración de la red, la calidad de la energía, la gestión de reservas, el manejo de la saturación y la previsión precisa.

Los estudios de integración de la red para la energía eólica han revelado muchos desafíos técnicos y socioculturales. El factor principal que determina una buena integración en la red de energía eólica es la capacidad de generación de los parques eólicos. Para los parques eólicos terrestres, la proximidad a la red eléctrica ha demostrado ser un factor beneficioso que, en última instancia, reduce los costos de transmisión y auxiliares. Una calendarización óptima en la generación de energía para un parque eólico implica pronósticos precisos y confiables. En términos de flexibilidad, la energía eólica se enfrenta a enormes desafíos, por ejemplo un cambio repentino en la magnitud y dirección de la velocidad del viento. Este fenómeno se denomina evento de rampa. Tales eventos pueden manejarse mejor si los pronósticos de energía eólica se hacen con alta precisión. La predicción de la energía eólica se inició a principios de la década de 1980 para modelar la velocidad del viento como series de tiempo [4]. Brown et al., han estudiado los modelos de series de tiempo esencialmente para simular la velocidad y la potencia del viento. Los pronósticos de velocidad del viento se transforman en pronósticos de energía eólica mediante una ley de transformación. Primero, la velocidad del viento se pronostica a una altura de base y luego se extrapola a la altura del cubo de la turbina utilizando la ley logarítmica o la ley de potencia. Se han aplicado modelos de series de tiempo como los modelos de media móvil autorregresiva (ARMA en inglés Autoregressive Moving Average) y de media móvil integrada autorregresiva (ARIMA en inglés Autoregressive Integrated Moving Average) para pronosticar la velocidad del viento y planificar el despacho de energía eólica para un funcionamiento óptimo del sistema de energía. Sin embargo, la precisión obtenida con estos modelos es baja debido a varias limitaciones planteadas por la naturaleza no lineal y estocástica de la velocidad del viento [1].

El año 2020 fue el mejor de la historia para la industria eólica mundial, con un crecimiento interanual del 53 %. La instalación de más de 93 GW de energía eólica en un año ha sido desafiante, con interrupciones tanto en la cadena de suministro global como en la construcción del proyecto, ha demostrado la increíble capacidad de recuperación de la industria eólica.

Los 93 GW de nuevas instalaciones elevan la capacidad global acumulada de energía eólica hasta 743 GW. En el mercado onshore, se instalaron 86,9 GW, un aumento del 59 % en comparación con 2019. China y los Estados Unidos siguieron siendo los mercados más grandes del mundo para nuevas incorporaciones onshore, y las dos principales economías del mundo juntas aumentaron su participación de mercado entre un 15 % y 76 %. A nivel regional, 2020 también fue un año récord para las instalaciones en tierra en Asia Pacífico, América del Norte y América Latina. Las tres regiones combinadas instalaron un total de 74 GW de nueva capacidad eólica terrestre el año pasado, o un 76 % más que el año anterior [5]. La generación de energía eólica terrestre ha alcanzado la madurez tecnológica de ser competitiva con las opciones de generación de energía de menor costo en muchos lugares. Por ejemplo, en México en 2012 la capacidad instalada aumentó en un 76 % con respecto a la capacidad total instalada

de generación de energía eólica a fines de 2011, debido a la creciente explotación del intenso recurso en el estado de Oaxaca.

En Oaxaca en el corredor de la Venta a la Mata pasando por la Ventosa, la velocidad del viento promedio anual es superior a 9 m/s a 30 m sobre el nivel del suelo con una dirección del viento dominante de Norte-Noroeste/Norte-Noreste 70 % del tiempo [6]. Estas condiciones de vientos intensos altamente favorables en Oaxaca representan una fuente apreciable de energía renovable de bajo costo, además de las grandes reservas de combustibles fósiles de México, lo que hace que su explotación sea una prioridad. En México, el Centro Nacional de Control de Energía (CENACE) es responsable del control de despacho de energía para el Sistema Eléctrico Nacional. El CENACE utiliza un sistema de información para preparar estrategias previas al envío. Este sistema toma en cuenta: disponibilidad, derrateo, restricciones y otros factores que afectan la capacidad de despacho de las unidades generadoras, así como la previsión de demanda eléctrica. Recientemente, se ha desarrollado un número considerable de modelos de predicción de la velocidad del viento utilizando una variedad de métodos, algunos simples y otros que combinan varias técnicas [7].

1.2. Motivación y planteamiento del problema

Recientemente, el uso de energías limpias ha crecido a gran escala en los últimos años, aun así, los combustibles fósiles no dejan de ser relevantes para el uso de la vida diaria, casi el 86 % de fuentes de energía son fósiles, lo que se busca cambiar para el futuro es reducir ese porcentaje usando energías limpias como la eólica. Aunque en México se cuenta con alrededor 68 parques eólicos aproximadamente, no existen una buena cantidad de artículos y proyectos destinados a la predicción de energía eólica en el país. El obtener y acceder a esta información puede ayudar tanto a calcular la generación de energía eólica, así como, obtener información respecto a la vida útil de los generadores y a gran escala conocer la factibilidad de crear un parque eólico en una zona geográfica predeterminada del país.

Se han investigado modelos basados en IA (Inteligencia Artificial) para la predicción del viento. [8] y [9] usan aprendizaje por reforzamiento, [10] y [11] usan lógica difusa, [12], [13], [14], y [15] utilizan SVM (del inglés support vector machine). Pasari et. al. [16] tuvo en cuenta la función de activación de la memoria a largo plazo, la unidad lineal rectificada y el algoritmo de optimización de Adam, para realizar un modelo de predicción de la velocidad del viento basado en redes neuronales artificiales. Navas et. al. [17] utilizó diferentes redes neuronales, como el modelo de red neuronal de multicapa-perceptron, la red neuronal de función de base radial y la regresión categórica, para predecir con precisión la velocidad del viento. Amir et. al. [18] construyó un modelo basado en redes neuronales artificiales, que mide la velocidad del viento con mayor precisión, utilizando algoritmos de entrenamiento, regularización bayesiana, Levenberg Marquardt y gradiente conjugado escalado. Madhiarasan [19] realizó una red neuronal de función de base radial recursiva utilizando tres variables,

que son la dirección del viento, la temperatura y la velocidad del viento para mejorar los sistemas eléctricos. Navas [17] y Madhiarasa [19] utilizaron redes neuronales para mejorar la precisión de la predicción.

Sin embargo, los métodos anteriores no funcionan bien cuando los conjuntos de datos de energía eólica tienen muestras pequeñas o faltan datos, ya que los métodos basados en estadística y aprendizaje automático necesitan grandes datos para entrenar los modelos. Para mejorar la predicción, especialmente a largo plazo con muestras pequeñas, utilizamos el método de multimodelos y redes neuronales para ARIMA. Usaremos la técnica de transferencia de aprendizaje para expandir los datos de entrenamiento con los conjuntos de datos de otros parques eólicos. También combinaremos varios modelos de parques eólicos con redes neuronales. La idea básica es usar el conocimiento de diferentes lugares de monitoreo para evitar el problema de entrenamiento causado por muestras pequeñas.

1.3. Objetivo y contribución de la tesis

La contribución objetivo fue desarrollar un modelo para la predicción de energía eólica con conjuntos de datos de serie de tiempo, y obtener una mejora de predicción que supere los estándares del modelo ARIMA. En esta tesis se construyó un nuevo modelo que se basa en utilizar conjuntos de datos de 3 diferentes granjas eólicas con distintas localizaciones del mundo. Con el fin de obtener diversas características y obtener un modelo predictivo. Esta propuesta no se limita a la combinación de un solo modelo ARIMA con red neuronal, como lo proponen [[20], [21]]. La diferencia radica en desarrollar una estrategia multimodelo; que se refiere a construir modelos ARIMA óptimos para cada conjunto de datos, entrenar tales modelos y posteriormente unirlos mediante una red neuronal. Entre la unión de los modelos se utiliza como base el principio de aprendizaje por transferencia; en el que desea construir un dominio más grande para mejorar características de un modelo predictivo. Con las pruebas realizadas se comprueba que el modelo funciona para diversos conjuntos, en este caso los tres propuestos; mejora la precisión de cada modelo respecto a un modelo solamente basado en ARIMA; aporta una mejora significativa a un modelo creado con pocos datos, y puede ser aplicado en su mayoría a datos basados en series de tiempo que constan de una sola variable. Concluyendo que el modelo propuesto obtiene buena precisión en las predicciones según las métricas de medición de errores.

1.4. Organización de la tesis

La organización del documento está seccionada de la siguiente manera. En el capítulo 1 se describe de modo resumido la energía eólica, así como los tipos de clasificación que existen para la velocidad y la energía generada por viento. Se analiza la energía en forma de serie de tiempo y se da una definición de esta, para después abordar el comportamiento más básico de una serie en su forma estacionaria. En el capítulo 2 plantea los tipos de modelado para la predicción de series de tiempo, enfocándose en la forma lineal de una serie estacionaria. El capítulo 3 continua los tipos de modelos para series con enfoque a las no estacionarias e introduce a los modelos múltiples. El capítulo 4 se centra en modelos no lineales de predicción, específicamente las redes neuronales que son fundamentales para el modelo planteado, y las formas de medir el error de una predicción a través de las métricas de estimación. El capítulo 5 es la concentración del estado del arte, posteriormente se realiza un contraste con el modelo planteado, que propone resolver la predicción eólica mediante la estrategia de multimodelos ARIMA. El capítulo 6 detalla la aplicación del modelo múltiple ARIMA y los resultados que se obtuvieron al ejecutar este modelo. Por último, el capítulo 7 describe las conclusiones del trabajo y los trabajos a futuro.

Capítulo 2

Fundamentos

El viento es una fuente de energía que ha sido utilizada en variadas aplicaciones. En la actualidad, el uso de la energía eólica ha evolucionado principalmente en la generación de electricidad. En este capítulo se dará una introducción al viento como una variable física estocástica, se describirá una vista general de los mecanismos y métodos que existen para predecir la energía eólica en estos últimos años. Por último, analizaremos el viento como única variable respecto al tiempo, para conocer a detalle el concepto de series de tiempo e introducir los tipos de serie estacionaria y no estacionaria.

2.1. Energía eólica

El viento es aire atmosférico en movimiento. Es omnipresente y uno de los elementos físicos básicos de nuestro entorno. Dependiendo de la velocidad del aire en movimiento, el viento puede sentirse ligero y etéreo, siendo silencioso e invisible a simple vista. O bien, puede ser una fuerza fuerte y destructiva, ruidosa y visible como resultado de los pesados escombros que arrastra. La velocidad del movimiento del aire define la fuerza del viento y está directamente relacionada con la cantidad de energía del viento, es decir, su energía cinética. La fuente de esta energía, sin embargo, es la radiación solar. La radiación electromagnética del sol calienta la superficie de la Tierra de manera desigual, más fuerte en los trópicos y más débil en las latitudes altas. Además, como resultado de una absorción diferencial de la luz solar por parte del suelo, las rocas, el agua y la vegetación, el aire en diferentes regiones se calienta a un ritmo diferente. Este calentamiento desigual se convierte a través de procesos convectivos en movimiento de aire, que se ajusta por la rotación de la Tierra. Los procesos convectivos son perturbaciones del equilibrio hidrostático mediante el cual, de lo contrario, las masas de aire estancadas se desplazan y se mueven en reacción a las fuerzas inducidas por los cambios en la densidad del aire y la flotabilidad debido a las diferencias de temperatura. El aire es empujado desde regiones de alta a baja presión, equilibrando la fricción y las fuerzas de inercia debidas a la rotación de la Tierra [22].

Los patrones de calentamiento diferencial de la superficie de la Tierra, así como otros procesos térmicos como la evaporación, la precipitación, las nubes, la sombra y las variaciones de la absorción de la radiación superficial, aparecen en diferentes escalas de espacio y tiempo. Estos se combinan con las fuerzas dinámicas debidas a la rotación de la Tierra y la redistribución del impulso del flujo para impulsar una variedad de procesos de generación de viento, lo que lleva a la existencia de una gran variedad de fenómenos eólicos. Estos vientos se pueden categorizar en función de su escala espacial y mecanismos físicos de generación [22].

2.2. Predicción del viento: descripción general

Con el viento disponible en forma abundante, aprovechar la energía del viento es una tarea especializada. El procesamiento de errores de la potencia y/o velocidad del viento pronosticada y la potencia y/o velocidad del viento real desempeñan un papel fundamental en la selección de un algoritmo de predicción aplicable de forma adecuada. El pronóstico del viento juega un papel importante cuando se trata de despejar los escenarios del mercado del día siguiente. Dado que existe una situación de mercado que aclarar, un esquema de pronóstico de viento preciso es útil en tales situaciones. La clasificación no solo simplifica el estudio, sino que también ayuda a elegir el método correcto según el tipo de su aplicación [1].

Se han propuesto e implementado distintos enfoques para el pronóstico de la generación de energía eólica. De acuerdo con diferentes estándares de clasificación se pueden dividir en los grupos que muestra la figura 2.1. A continuación, se dará una breve descripción de cada categoría como ejemplos remarcados en la literatura.

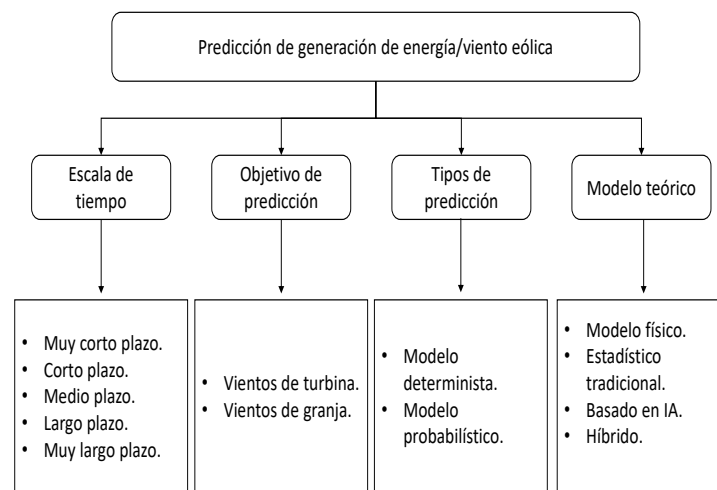


Figura 2.1: Tipos de clasificación de predicción de energía eólica.

Por escala de tiempo

En términos de la preferencia de la industria, se han desarrollado muchas herramientas de pronóstico del viento basadas en el horizonte de predicción. Las etapas de pronóstico del viento se pueden clasificar en:

- Predicción a muy corto plazo (de unos segundos a 30 minutos)

- Predicción a corto plazo (30 minutos a 6 horas)
- Predicción a medio plazo (de 6 a 24 horas)
- Predicción a largo plazo (24 horas a 72 horas)
- Predicción a muy largo plazo (72 horas y más)

La previsión a corto plazo es una opción preferida debido a las operaciones del mercado y al óptimo despacho de energía. En estos días, los mercados de electricidad a menudo siguen una estructura desregulada que permite a varios agentes de licitación competidores participar en el proceso minorista. Dado que la energía producida a partir de recursos eólicos es de naturaleza estocástica, a menudo se considera que un mercado eléctrico diario planifica el despacho óptimo. Este despacho depende esencialmente de los programas de energía eólica proporcionados por el operador del mercado. Una producción de energía eólica estocástica trae errores en las grandes previsiones, lo que exige la necesidad de capacidad de reserva para compensar las potencias deficitarias. Si no se dispone de suficientes reservas, el costo en el que se incurre para el operador del mercado aumenta drásticamente. Dado que la penetración de la energía eólica aumenta significativamente a diario, se requieren esquemas de pronóstico más eficientes para garantizar el equilibrio en los mercados de la electricidad [1].

Por objetivo de predicción

De acuerdo con los diferentes objetivos de pronóstico, los modelos también se pueden clasificar en dos tipos: vientos de turbina y vientos de granja eólica para predicción de WS (del inglés Wind Speed) y/o WP (del inglés Wind Power). El primero puede ayudar a caracterizar la potencia de salida de una sola turbina eólica [[23], [24], [25]]. Este último integra información de muchas turbinas eólicas para pronosticar la producción de energía de un parque eólico completo [[26], [27], [28]]. De los dos, el último enfoque es más difícil[29].

Por tipo de predicción

Según el tipo de pronóstico proporcionado, los modelos WS/WP se pueden dividir en modelos deterministas y probabilísticos [30]. Los modelos deterministas, que solo brindan pronósticos WS/WP puntuales [[31], [32], [33], [34]], tienen un rendimiento de pronóstico limitado debido a la complejidad del entorno y, a veces, generan pronósticos insatisfactorios con muchos errores. Los modelos probabilísticos, que a menudo brindan más información para los tomadores de decisiones que los pronósticos puntuales [31], pueden representar la incertidumbre en términos de intervalos de predicción (PI en inglés Interval Prediction) [[35], [31], [33]–[36]] [29].

Por modelo teórico

Los modelos de pronóstico también se pueden dividir en cuatro tipos según las diferencias en la teoría de modelos [[34], [37]]: modelos físicos, modelos estadísticos tradicionales, modelos basados en inteligencia artificial (AI del inglés Artificial Intelligence) y modelos híbridos. Los modelos físicos, como la predicción numérica del tiempo (NWP del inglés Numerical Weather Prediction) y el pronóstico meteorológico (WRF del inglés Weather Research and Forecasting), generalmente consideran varios factores meteorológicos (p. ej., presión del aire, humedad y temperatura) para pronosticar el futuro WS [[38], [34]]. Estudios previos mostraron que los modelos físicos tenían un mejor desempeño en el pronóstico de WS a mediano y largo plazo [34], lo que permitió obtener pronósticos de WP utilizando la curva de potencia de un aerogenerador en un WS dado [39]. Los modelos estadísticos tradicionales incluyen el autorregresivo de media móvil (ARMA) [40], el autorregresivo integrado de media móvil (ARIMA) [41], ARIMA fraccional (f-ARIMA) [42] y el modelo autorregresivo de Hammerstein [43]. A diferencia de los modelos físicos, estos son modelos de series de tiempo que generalmente se emplean para caracterizar la fluctuación lineal de WS/WP en diferentes ubicaciones y, en general, muestran un buen rendimiento en el pronóstico de WS/WP a muy corto plazo y a corto plazo [38]. Con los avances en informática, muchos modelos basados en IA también se han utilizado ampliamente en el pronóstico de WS/WP, por ejemplo, la máquina de vectores de soporte (SVM del inglés Support Vector Machine) [44], la máquina de vectores de soporte de mínimos cuadrados (LSSVM del inglés Least-Squares Support Vector Machine) [45], la máquina de aprendizaje extremo (ELM del inglés Extreme Learning Machine) [46], kernel ELM (KELM del inglés Kernel Extreme Learning Machine) [47], método de lógica difusa [48], filtro de Kalman [49] y diferentes tipos de redes neuronales artificiales (ANN del inglés Artificial Neural Networks), incluida la red neuronal de propagación inversa (BPNN del inglés Backpropagation Artificial Neural Network) [50], perceptrón multicapa (MLP del inglés Multi-Layer Perceptron) [51], red neuronal wavelet (WNN de inglés Wavelet Neuronal Network) [50], red Elman [52] y red neuronal de regresión generalizada (GRNN del inglés General Regression Neuronal Network) [50]. En la actualidad, con el desarrollo de la tecnología de aprendizaje profundo, las redes neuronales profundas (DNN del inglés Deep Neuronal Network), incluida la memoria a corto plazo (LSTM del inglés Long Short-Term Memory) [46], la red neuronal convolucional (CNN del inglés Convolutional Neuronal Network) [35], la LSTM bidireccional (BiLSTM) [53], unidad recurrente cerrada (GRU del inglés Gated Recurrent Unit) [54], GRU bidireccional (BiGRU) [55], red de creencia profunda (DBN del inglés Deep Belief Network) [56] y codificador automático (AE del inglés Automatic Encryption) [57], han atraído una mayor atención en el pronóstico WS/WP debido a su capacidad superior para tratar problemas no lineales complejos. En muchos estudios anteriores, los modelos basados en IA mostraron un mejor rendimiento de pronóstico de WS/WP que los modelos estadísticos tradicionales. Se pueden desarrollar modelos híbridos combinando diferentes tipos de modelos para caracterizar diferentes aspectos de la fluctuación WS/WP [58], [29].

2.3. Series de tiempo

Pronosticar una variable en particular puede depender de la escala temporal o espacial. Las variaciones temporales con el tiempo reflejan la estocasticidad presente en la variable. Una escala temporal para una variable se puede modelar en términos de series de tiempo.

Una serie de tiempo es el resultado de observar los valores de una variable a lo largo del tiempo en intervalos regulares (cada hora, cada día, cada mes, entre otros). Matemáticamente está compuesto por variables aleatorias en una secuencia ordenada con respecto al tiempo, que suele ser llamada un proceso estocástico. En un principio las series se modelaban por las 4 principales componentes: tendencia, estacional, aleatoria y cíclica, más adelante se explican con detalle.

Definición de una serie de tiempo

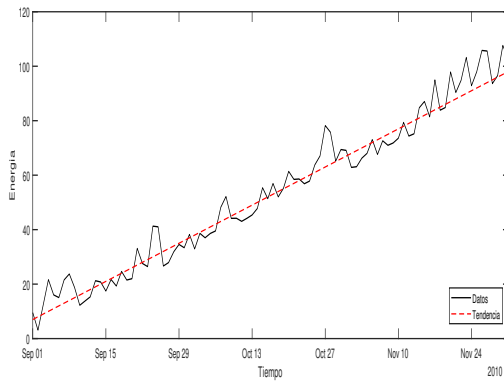
Una serie de tiempo es un conjunto de observaciones generadas secuencialmente a lo largo del tiempo. Si el conjunto es continuo, la serie temporal es continua. Si el conjunto es discreto, entonces la serie temporal es discreta. Por lo tanto, las observaciones de una serie de tiempo discreta realizadas en los tiempos $\tau_1, \tau_2, \dots, \tau_t, \tau_N$ pueden denotarse por $z(\tau_1), z(\tau_2), \dots, z(\tau_t), z(\tau_N)$. Con este contexto, consideramos solo series de tiempo discretas, donde las observaciones se realizan en un intervalo fijo h . Cuando tenemos N valores sucesivos de dicha serie disponibles para el análisis, escribimos $z_1, z_2, \dots, z_t, z_n$ para denotar observaciones realizadas en intervalos de tiempo equidistantes $\tau_0 + h, \tau_0 + 2h, \dots, \tau_0 + th, \dots, \tau_0 + Nh$. Si nombramos τ_0 como origen y h como unidad de tiempo, podemos considerar z_t como la observación en el tiempo t [59].



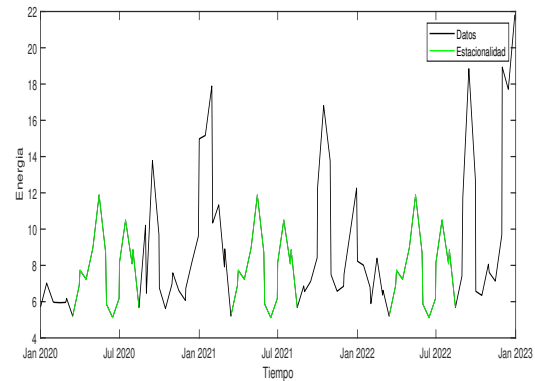
Figura 2.2: Ejemplo de una serie de tiempo, donde z_t es el rendimiento de 100 repeticiones de un proceso aleatorio.

Componentes de una serie de tiempo

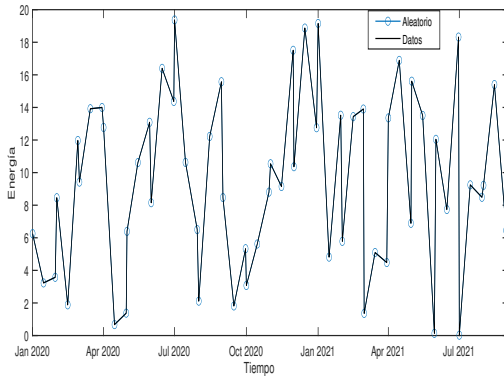
Las series de tiempo se componen en cuatro características fundamentales: tendencia, estacional, aleatoria y cíclica. La tendencia en una serie es el cambio a largo plazo que produce en la relación de la media de los datos. El componente estacional es cuando una serie presenta cierta periodicidad, variación de cierto periodo ya sea semestral, mensual, además de otros. La componente aleatoria no responde a ningún patrón de comportamiento, básicamente es el resultado de un factor aleatorio que incide de forma aislada en una serie de tiempo. Y por último la componente cíclica que, aunque su duración es irregular, son patrones en la serie que se identifican en ciertos intervalos de tiempo. Cabe aclarar que para diferenciar la componente estacional de la cíclica debemos tener en cuenta que la estacional ocurre en intervalos de tiempo conocidos mientras que la cíclica no se puede determinar. En la figura 2.3 se muestra el comportamiento de cada componente en distintas series de tiempo.



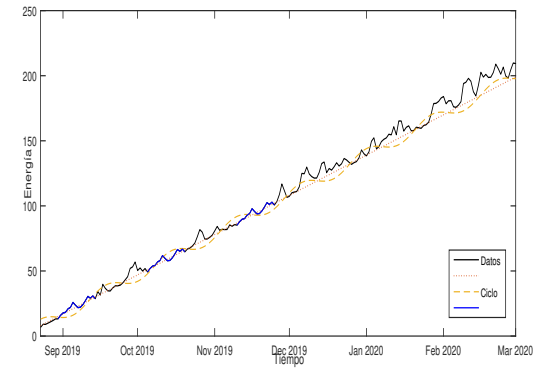
(a) Tendencia



(b) Estacional



(c) Aleatoria



(d) Cíclica

Figura 2.3: Componentes de una serie de tiempo.

Clasificación de las series de tiempo

Los fenómenos dinámicos que desatan las series temporales pueden clasificarse en dos tipos de clases:

- Estacionarias.

Una serie es estacionaria cuando los valores de esta son estables en un nivel constante sin mostrar una tendencia a crecer o decrecer a largo plazo, es decir la media y la varianza son constantes en el tiempo.

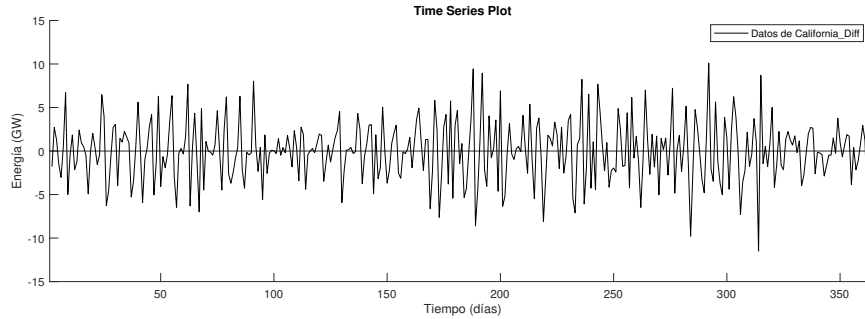
Matemáticamente una serie estacionaria S_t puede expresarse como:

$$F_S(S_{t_1+k}, \dots, S_{t_n+k}) = F_S(S_{t_1}, \dots, S_{t_n}) \quad (2.1)$$

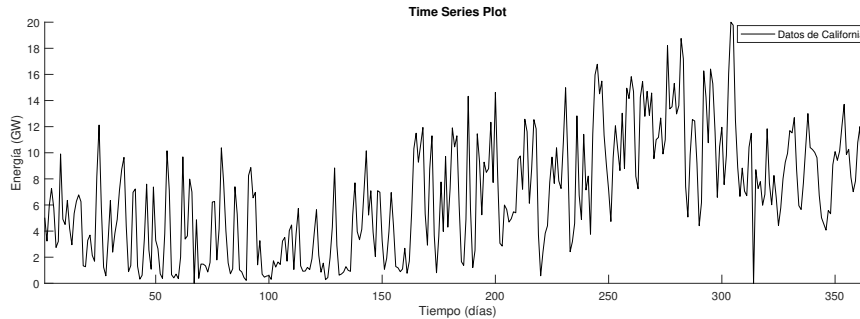
Donde t es el paso del tiempo y k es el paso del tiempo desplazado [1]. La figura 2.4(a) da un ejemplo claro de una serie estacionaria.

- No estacionarias.

Una serie no es estacionaria cuando las tendencias y/o variabilidad cambia con el tiempo. Los cambios en la media determinan tendencia a crecer o decrecer a largo plazo, por lo que la serie no oscila alrededor de un valor constante. En la figura 2.4(b) muestra el ejemplo de una serie no estacionaria.



(a) Serie estacionaria



(b) Serie no estacionaria

Figura 2.4: Series de tiempo de datos adquiridos de los parques eólicos de California.

2.4. Análisis de series estacionarias

Previamente vimos que existen dos clasificaciones para las series de tiempo, estacionarias y no estacionarias; en esta sección nos enfocaremos en el tipo de serie estacionaria, el método ADF (del inglés Augmented Dickey-Fuller) para comprobar la clasificación de estacionaria, la definición de la serie vista como un proceso estocástico, el comportamiento respecto a la varianza y su correlación en una serie de tiempo.

2.4.1. Comprobación de serie estacionaria (Prueba Dickey-Fuller Aumentada)

La prueba de Dickey-Fuller aumentada es la versión extendida de la prueba simple de Dickey-Fuller. Debido a que es poco probable que el término de error sea ruido blanco, ampliaron la prueba para incluir rezagos adicionales en términos de las variables dependientes, para eliminar el problema de la autocorrelación. Normalmente se utiliza la prueba de Dickey-Fuller aumentada en lugar de la prueba de Dickey-Fuller simple. La diferencia de ambas pruebas es que se incluyen los valores rezagados de la variable dependiente en el modelo simple, y se continua este procedimiento hasta que se elimine la autocorrelación. Esto puede ilustrarse como

$$\begin{aligned} y_t &= \beta_1 + \beta_2 y_t + \epsilon_t \\ y_t &= \beta_1 + \beta_2 y_t + \beta_3 y_{t-1} + \epsilon_t \\ y_t &= \beta_1 + \beta_2 y_t + \beta_3 y_{t-1} + \beta_4 y_{t-2} + \epsilon_t \end{aligned} \tag{2.2}$$

Ahora

$$\begin{aligned} \Delta y_t &= \gamma y_{t-1} + \beta_1 \Delta y_{t-1} + \epsilon_t \\ \Delta y_t &= \gamma y_{t-1} + \beta_1 \Delta y_{t-1} + \beta_2 \Delta y_{t-2} + \cdots + \beta_p \Delta y_{t-p} + \epsilon_t \end{aligned} \tag{2.3}$$

El proceso continúa hasta que se elimine la autocorrelación. Esta expresión se denota como:

$$\begin{aligned} \Delta y_t &= \gamma y_{t-1} + \sum_{i=1}^p \beta_i \Delta y_{t-i} + \epsilon_t \\ \Delta y_t &= \alpha + \gamma y_{t-1} + \sum_{i=1}^p \beta_i \Delta y_{t-i} + \epsilon_t \\ \Delta y_t &= \alpha + \beta_t + \gamma y_{t-1} - 1 + \sum_{i=1}^p \beta_i \Delta y_{t-i} + \beta_2 \Delta y_{t-2} + \cdots + \beta_p \Delta y_{t-p} + \epsilon_t \end{aligned} \tag{2.4}$$

Algunas suposiciones comunes de mínimos cuadrados ordinarios (OLS) se discuten aquí:

1. ϵ debe ser independiente.

2. No debe haber heterocedasticidad, debe homogeneidad.
3. No debería haber ruptura estructural, el coeficiente debería ser estable.
4. El término de error debe tener una distribución normal.

La prueba de estacionariedad en la prueba de Dickey-Fuller aumentada sigue el mismo procedimiento que en la prueba de Dickey-Fuller simple. Primero, estacionario se verifica al nivel que en primera diferencia y finalmente en segunda diferencia. A primera diferencia la ecuación será la siguiente:

$$\begin{aligned}\Delta_2 y_t &= \gamma y_{t-1} + \sum_{i=1}^p \beta_1 \Delta_2 y_{t-1} + \epsilon_t \\ \Delta_2 y_t &= \alpha + \gamma y_{t-1} + \sum_{i=1}^p \beta_1 \Delta_2 y_{t-1} + \epsilon_t \\ \Delta_2 y_t &= \alpha + \beta_t + \gamma y_{t-1} - 1 + \sum_{i=1}^p \beta_1 \Delta_2 y_{t-1} + \beta_2 \Delta_2 y_{t-2} + \cdots + \beta_p \Delta_2 y_{t-p} + \epsilon_t\end{aligned}\tag{2.5}$$

Si la serie aún no es estacionaria, se utilizan las mismas ecuaciones reemplazando Δ_2 con Δ_3 el proceso será el mismo siempre y cuando la serie no se vuelva estacionaria.

Prueba de Hipótesis

$$\begin{aligned}H_0 : \gamma &= 0 \\ H_1 : \gamma &< 0\end{aligned}\tag{2.6}$$

La hipótesis nula se prueba a través de la estadística t que viene dada por esta fórmula:

$$t = \frac{\hat{\gamma} - \gamma H_0}{SE(\hat{\gamma})}\tag{2.7}$$

Si t calculado es mayor que el valor crítico no rechazamos nuestra hipótesis nula. En esta situación la variable en consideración será no estacionaria y tendrá raíz unitaria. Por otro lado, si t calculado es menor que el valor crítico, rechazamos nuestra hipótesis nula. En este caso la serie subyacente sería una serie estacionaria y no tiene raíz unitaria. Primero, la serie se prueba en nivel normal y en caso de que no se vuelve estacionaria; se procede a realizar la prueba con la serie en primera y segunda diferencia secuencialmente. Hay otro método para rechazar o no rechazar la hipótesis nula; si el valor calculado está en el lado derecho del valor crítico, en la cola de un lado (ver figura 2.5) no rechazamos la hipótesis nula, y si el valor calculado está en el lado izquierdo del valor crítico, rechazamos la hipótesis nula y concluimos que la serie no tiene raíz unitaria. El valor p también se usa para rechazar o aceptar la hipótesis nula si el valor $p < .05$ rechaza la hipótesis nula y viceversa [60].

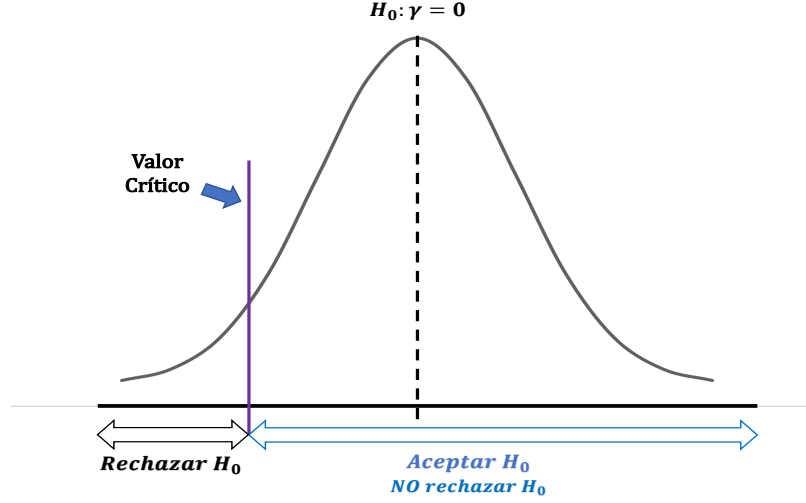


Figura 2.5: Gráfica para Dickey-Fuller aumentado

2.4.2. Proceso estocástico

Un fenómeno estadístico que evoluciona en el tiempo de acuerdo con leyes probabilísticas se denomina proceso estocástico. A menudo nos referiremos a él simplemente como un proceso, omitiendo la palabra “estocástico”. Las series de tiempo pueden pensarse entonces como una realización particular, producida por el mecanismo de probabilidad subyacente, del sistema en estudio. En otras palabras, al analizar una serie de tiempo la consideramos como la realización de un proceso estocástico [59].

Proceso estacionario estocástico

Una clase muy especial de procesos estocásticos, llamados procesos estacionarios, se basa en la suposición de que el proceso se encuentra en un estado particular de equilibrio estadístico. Se dice que un proceso estocástico es estrictamente estacionario si sus propiedades no se ven afectadas por un cambio en el origen temporal, es decir, si la distribución de probabilidad conjunta asociada con m observaciones $z_{t_1}, z_{t_2}, \dots, z_{t_m}$, realizada en cualquier conjunto de tiempos t_1, t_2, \dots, t_m , es el mismo que el asociado a m observaciones $z_{t_1+k}, z_{t_2+k}, \dots, z_{t_m+k}$, realizadas en los momentos $t_1+k, t_2+k, \dots, t_m+k$. Por lo tanto, para que un proceso discreto sea estrictamente estacionario, la distribución conjunta de cualquier conjunto de observaciones no debe verse afectada por el desplazamiento de todos los tiempos de observación hacia adelante o hacia atrás en cualquier número entero k [59].

Media y varianza de un proceso estacionario

Cuando $m = 1$, la suposición de proceso estacionario implica que la distribución de probabilidad $p(z_t)$ la misma para todo t y puede escribirse como $p(z)$. Por lo tanto, el

proceso estocástico tiene una media constante

$$\mu E[z_t] = \int_{-\infty}^{\infty} zp(z)dz \quad (2.8)$$

que define el nivel sobre el cuál oscila, y tiene una varianza constante

$$\sigma = E[z_t - \mu]^2 = \int_{-\infty}^{\infty} (z - \mu)^2 p(z) dz \quad (2.9)$$

que mide su dispersión sobre este nivel. Dado que la distribución de probabilidad $p(z)$ es la misma para todos los tiempos t , su forma se puede inferir formando el histograma de las observaciones $\{z_1, z_2, \dots, z_N\}$ que componen la serie de tiempo observada. Además, la media μ del proceso estocástico se puede estimar mediante la media muestral

$$\bar{z} = \frac{1}{N} \sum_{t=1}^N z_t \quad (2.10)$$

de la serie temporal, y la varianza σ_z^2 del proceso estocástico se puede estimar mediante la varianza muestral de la serie temporal [59].

$$\sigma_z^2 = \frac{1}{N} \sum_{t=1}^N (z_t - \bar{z})^2 \quad (2.11)$$

Coeficientes de Autocorrelación y Autocovarianza

La suposición de estacionariedad también implica que la distribución de probabilidad conjunta $p(z_{t_1}, z_{t_2})$ es la misma para todos los tiempos t_1, t_2 , que están separados por un intervalo constante. En particular, se sigue que la covarianza entre los valores z_t y z_{t+k} , separados por k intervalos de tiempo, o por rezago k , debe ser la misma para todo t bajo el supuesto de estacionariedad. Esta covarianza se llama autocovarianza en el rezago k y está definida por:

$$\gamma_k = \text{cov}[z_t, z_{t+k}] = E[(z_t - \mu)(z_{t+k} - \mu)] \quad (2.12)$$

Del mismo modo, la autocorrelación en el rezago k es:

$$\begin{aligned} \rho_k &= \frac{E[(z_t - \mu)(z_{t+k} - \mu)]}{\sqrt{E[(z_t - \mu)^2]E[(z_{t+k} - \mu)^2]}} \\ &= \frac{E[(z_t - \mu)(z_{t+k} - \mu)]}{\sigma_z^2} \end{aligned} \quad (2.13)$$

ya que, para un proceso estacionario, la varianza $\sigma_z^2 = \gamma_0$ es la misma en el tiempo $t + k$ que en el tiempo t . Por lo tanto, la autocorrelación en el rezago k , es decir, la correlación entre z_t y z_{t+k} , es

$$\rho = \frac{\gamma_k}{\gamma_0} \quad (2.14)$$

lo que implica, en particular, que $\rho_0 = 1$.

También se sigue para un proceso estacionario que la naturaleza de la distribución de probabilidad conjunta $\rho(z_t, z_{t+k})$ de valores separados por k intervalos de tiempo se puede inferir trazando un diagrama de dispersión usando pares de valores (z_t, z_{t+k}) de la serie de tiempo, separados por un intervalo constante o rezago k . Para los datos por lotes que se muestran en la figura 2.6, la figura 2.7a muestra un diagrama de dispersión para el rezago $k = 1$, obtenido al graficar z_{t+1} contra z_t , mientras que la figura 2.7b muestra un diagrama de dispersión para el rezago $k = 2$, obtenido trazando z_{t+2} contra z_t . Vemos que los valores vecinos de la serie de tiempo están correlacionados. La correlación entre z_t y z_{t+1} parece ser negativa y la correlación entre z_t y z_{t+2} positiva [59].

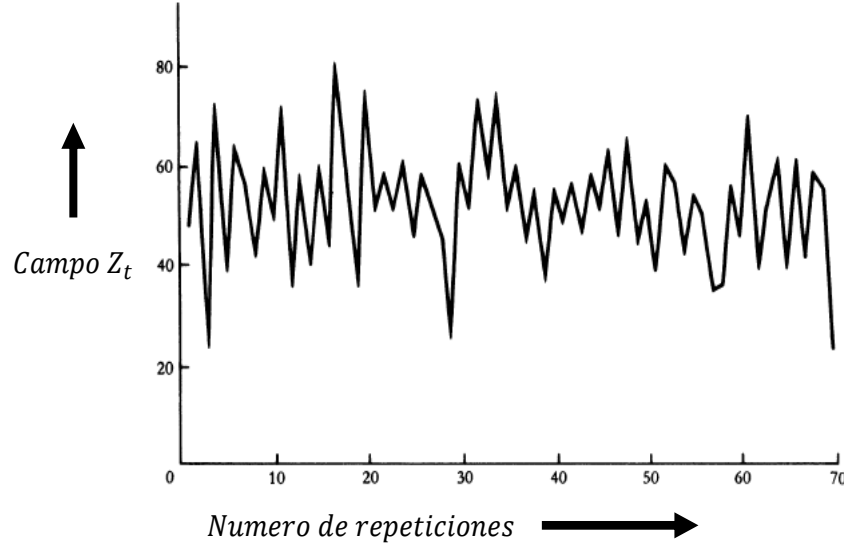


Figura 2.6: Campo z_t de 70 repeticiones de un proceso aleatorio.

Proceso de ruido blanco

El ejemplo fundamental de un proceso estacionario es una secuencia de variables aleatorias independientes e idénticamente distribuidas, indicadas como a_1, \dots, a_t, \dots , que también suponemos que tienen media cero y varianza σ_a^2 . Este proceso es estrictamente estacionario y se denomina proceso de ruido blanco. Como la independencia implica que las a_t no están correlacionadas, su función de autocovarianza es simplemente:

$$\gamma_k = E[a_t a_{t+k}] = \begin{cases} \sigma_a^2 & k = 0 \\ 0 & k \neq 0 \end{cases}$$

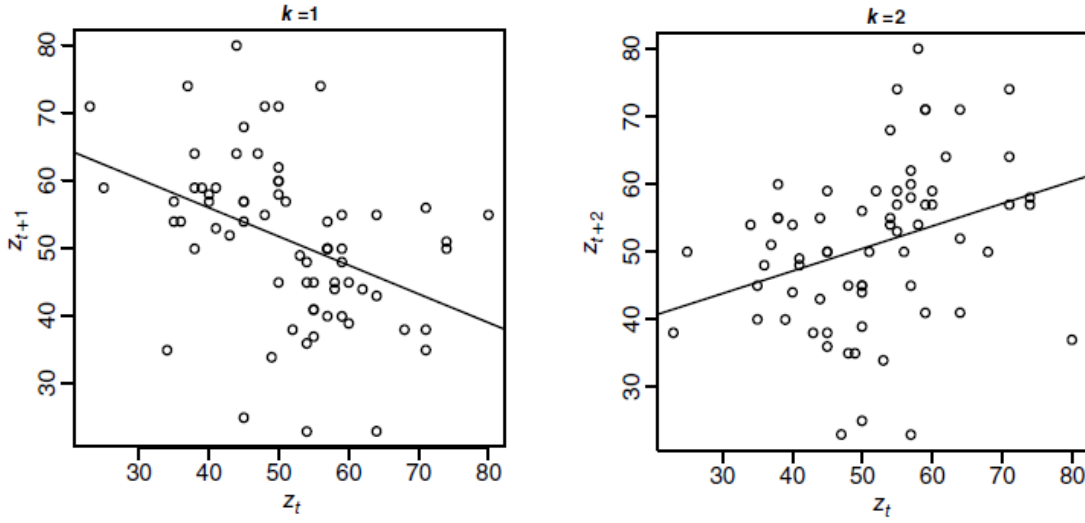
(a) $K = 1$ para datos de la figura 2.6(b) $K = 2$ para datos de la figura 2.6.

Figura 2.7: Diagramas de Scatter.

Si uno se concentra solo en las propiedades de segundo orden, entonces una secuencia de variables aleatorias a_t , que no están correlacionadas, tienen media cero y varianza constante (σ_a^2) tiene la misma función de autocovarianza γ_k que la anterior, y es débilmente estacionaria (de segundo orden). Tal proceso también puede denominarse proceso de ruido blanco (en el sentido débil), cuando el enfoque está solo en las propiedades de segundo orden. Aunque el proceso de ruido blanco tiene propiedades muy básicas, este proceso juega un papel importante en la construcción de procesos con propiedades mucho más interesantes y complicadas a través de operaciones de filtrado lineal [59].

2.4.3. Funciones de auto-covarianza y auto-correlación

Anteriormente se mostró que el coeficiente de autocovarianza y_k , con un rezago de k , mide la covarianza entre dos valores z_t y z_{t+k} separados por una distancia de k . La gráfica de y_k contra el rezago k se llama función de autocovarianza $\{y_k\}$ del proceso estocástico. De manera similar, la gráfica del coeficiente de autocorrelación ρ_k como función del rezago k se llama función de autocorrelación ρ_k del proceso. Tenga en cuenta que la función de autocorrelación es adimensional, es decir, independiente de la escala de medición de la serie temporal. Como $y_k = \rho_k \sigma_z^2$, el conocimiento de la función de autocorrelación $\{\rho_k\}$ y la varianza σ_z^2 es equivalente al conocimiento de la función de autocovarianza y_k [59].

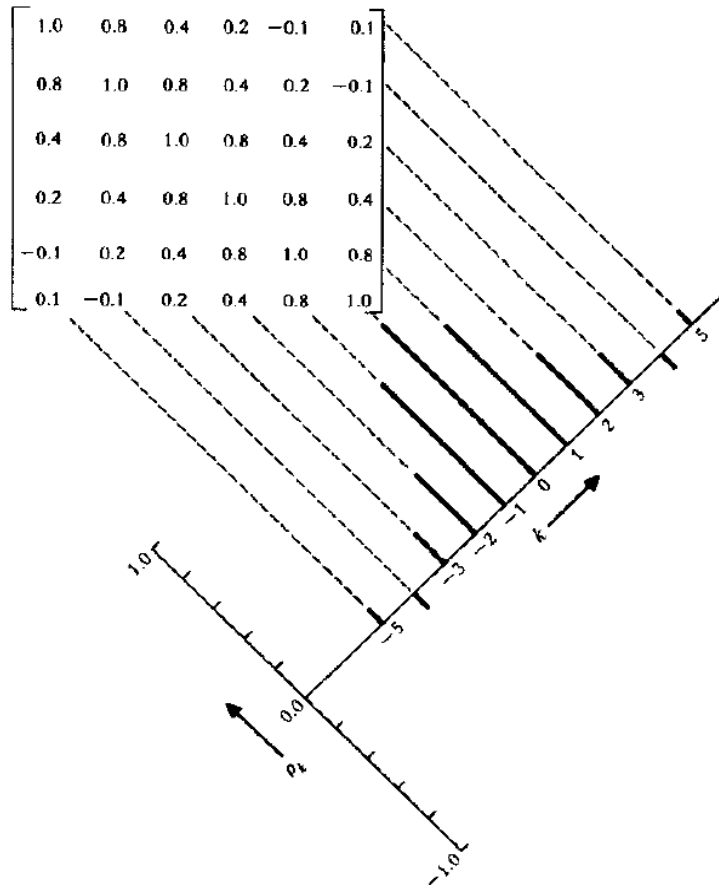


Figura 2.8: Matriz de autocorrelación y correspondencia de función de autocorrelación para proceso estacionario.

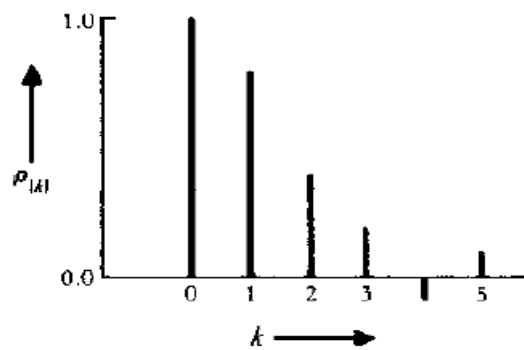


Figura 2.9: Función de autocorrelación para mitad positiva de la matriz.

La función de autocorrelación, que se muestra en la figura 2.8 como un gráfico de las diagonales de la matriz de autocorrelación, revela cómo cambia la correlación

entre dos valores cualquiera de la serie a medida que cambia su separación. Como $\rho_k = \rho_{-k}$, la función de autocorrelación es necesariamente simétrica con respecto a cero, y en la práctica solo es necesario graficar la mitad positiva de esta función. La figura 2.9 muestra la mitad positiva de la función de autocorrelación dada en la figura 2.8. Cuando hablamos de la función de autocorrelación, normalmente nos referimos a la mitad positiva. En el pasado, la función de autocorrelación a veces se llamaba correlograma. Por lo que se ha mostrado anteriormente, un proceso estacionario normal z_t está completamente caracterizado por su media μ y su función de autocovarianza γ_k , o de manera equivalente por su media μ , varianza σ_z^2 y función de autocorrelación ρ_k [59].

Capítulo 3

Modelos para predicción de series de tiempo

En la actualidad existen distintos modelos de predicción para la energía eólica. En especial para este tipo de energía, surge una gran popularidad modelarlos con un método estadístico muy popular y estable como ARIMA, que tiene una mejor aplicación cuando los datos se presentan en una serie de tiempo. El siguiente capítulo introduce los modelos lineales para series estacionarias (Procesos AR, MA y ARMA). Aunque en contraste en las aplicaciones del mundo real es complicado que una serie de tiempo tenga un comportamiento no cambiante, por ello también se introducen los modelos lineales para series no estacionarias que se basan en un proceso de convertir dicha serie no estacionaria en estacionaria. Para concluir el capítulo se verá un modelado para series no estacionarias (ARIMA) visto de forma múltiple.

3.1. Modelo lineal para series de tiempo estacionarias

Una clase importante de modelos estocásticos para describir series de tiempo, que ha recibido mucha atención, comprende los llamados modelos estacionarios. Los modelos estacionarios asumen que el proceso permanece en equilibrio estadístico con propiedades probabilísticas que no cambian con el tiempo, en particular, variando alrededor de un nivel medio constante fijo y con varianza constante [59].

La representación de un proceso estocástico como la salida de un filtro lineal, cuya entrada es ruido blanco a_t , es decir,

$$\begin{aligned}\tilde{z}_t &= a_t + \Psi_1 a_{t-1} + \Psi_2 a_{t-2} + \cdots \\ &= a_t + \sum_{j=1}^{\infty} \Psi_j a_{t-j}\end{aligned}\tag{3.1}$$

Donde $z_t = z_t - \mu$ es la desviación del proceso de algún origen, o de su media, si el

proceso es estacionario. El proceso lineal general 3.1 nos permite representar z_t como una suma ponderada de valores presentes y pasados del proceso de “ruido blanco” a_t . El proceso de ruido blanco a_t puede considerarse como una serie de choques que impulsan el sistema. Consiste en una secuencia de variables aleatorias no correlacionadas con media cero y varianza constante, es decir

$$E[a_t] = 0 \quad \text{var}[a_t] = \sigma_a^2$$

Dado que se supone que las variables aleatorias a_t no están correlacionadas, se deduce que su función de autocovarianza es

$$\gamma_k = E[a_t a_{t+k}] = \begin{cases} \sigma_a^2 & k = 0 \\ 0 & k \neq 0 \end{cases} \quad (3.2)$$

Así, la función de autocorrelación del ruido blanco tiene una forma particularmente simple

$$\rho_k = \begin{cases} 1 & k = 0 \\ 0 & k \neq 0 \end{cases} \quad (3.3)$$

Un resultado fundamental en el desarrollo de procesos estacionarios es el de Wold (1938), quien estableció que cualquier proceso estacionario puramente no determinista de media cero z_t posee una representación lineal como en (3.1) con $\sum_{j=0}^{\infty} |\Psi_j| < \infty$. Las a_t no están correlacionadas con la varianza común σ_a^2 pero no necesitan ser independientes. Reservaremos el término procesos lineales para procesos \tilde{z}_t de la forma de (3.1) en los que las a_t son variables aleatorias independientes. Para que \tilde{z}_t definida por (3.1) represente un proceso estacionario válido, es necesario que los coeficientes Ψ_j sean absolutamente sumables, es decir, para $\sum_{j=0}^{\infty} |\Psi_j| < \infty$. Bajo condiciones adecuadas (Koopmans, 1974), z_t es también una suma ponderada de pasados \tilde{z}_t y un choque agregado a_t , es decir,

$$\begin{aligned} \tilde{z} &= \pi_1 \tilde{z}_{t-1} + \pi_2 \tilde{z}_{t-2} + \cdots + a_t \\ &= \sum_{j=1}^{\infty} \pi_j \tilde{z}_{t-j} + a_t \end{aligned} \quad (3.4)$$

En esta forma alternativa, la desviación actual \tilde{z}_t desde el nivel μ se puede considerar como una ‘regresión’ en desviaciones pasadas $\tilde{z}_{t-1}, \tilde{z}_{t-2}, \dots$ del proceso [59].

3.1.1. Procesos Autorregresivos AR

Consideremos el caso especial de la ecuación (3.4) en el que solo el primer término p de los pesos es distinto de cero. El modelo puede escribirse como:

$$\tilde{z}_t = \phi_1 \tilde{z}_{t-1} + \phi_2 \tilde{z}_{t-2} + \dots + \phi_p \tilde{z}_{t-p} + a_t \quad (3.5)$$

Donde los simbolos $\phi_1, \phi_2, \dots, \phi_p$ son el conjunto finito de parámetros de peso. El proceso resultante se denomina proceso autorregresivo de orden p , o de una forma

más concisa, proceso AR(p).
En particular los modelos AR(1) y AR(2).

$$\begin{aligned}\tilde{z}_t &= \phi_1 \tilde{z}_{t-1} + a_t \\ &= \phi_1 \tilde{z}_{t-1} + \phi_2 \tilde{z}_{t-2} + a_t\end{aligned}\tag{3.6}$$

El modelo AR(p) se puede escribir en la forma equivalente

$$(1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p) \tilde{z}_t = a_t\tag{3.7}$$

o de forma sintetizada

$$\phi(B) \tilde{z}_t = a_t\tag{3.8}$$

Esto implica que:

$$\tilde{z} = \frac{1}{\phi(B)} a_t = \phi^{-1}(B) a_t = \psi(B) a_t\tag{3.9}$$

Por lo tanto, el proceso autorregresivo se puede considerar como la salida \tilde{z}_t de un filtro lineal con una función de transferencia $\phi^{-1}B = \psi(B)$ cuando la entrada es un ruido blanco a_t [59].

3.1.2. Procesos de medias móviles MA

Tenemos que considerar el caso especial de la ecuación (3.1). cuando solo el primer término de q de los ψ pesos es distinto de cero. Por tanto, el proceso puede ser escrito tal como:

$$\tilde{z}_t = a_t - \theta_1 \tilde{z}_{t-1} - \theta_2 \tilde{z}_{t-2} - \dots - \theta_p \tilde{z}_{t-p}\tag{3.10}$$

En dónde ahora $-\theta_1, -\theta_2, \dots, -\theta_q$ son los símbolos para el conjunto finito de parámetros de peso. Este proceso se denomina promedio móvil de orden a , que a menudo se abrevia como MA(q). Los casos especiales de MA(1) y MA(2).

$$\begin{aligned}\tilde{z}_t &= a_t - \theta_1 a_{t-1} \\ &= a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2}\end{aligned}\tag{3.11}$$

Son importantes para los modelos basados en ruido blanco. Utilizando el operador de retroceso $Ba_t = a_{t-1}$, el modelo MA(q) se puede escribir en la forma equivalente como:

$$\tilde{z}_t = (1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_q B^q) a_t\tag{3.12}$$

o en una forma compacta

$$\tilde{Z}_T = \theta(B) a_t\tag{3.13}$$

Por lo tanto, el proceso de media móvil se puede considerar una salida \tilde{z}_t de un filtro lineal con función de transferencia $\theta(B)$ cuando la entrada es el ruido blanco a_t [59].

3.1.3. Procesos mixtos autoregresivos-promedio móvil

El proceso de promedio móvil finito.

$$\tilde{z}_t = a_t - \theta_1 a_{t-1} = (1 - \theta_1 B) a_t \quad |\theta_1| < 1 \quad (3.14)$$

también se puede escribir como un proceso autorregresivo infinito

$$\tilde{z}_t = -\theta_1 \tilde{z}_{t-1} - \theta_1^2 \tilde{z}_{t-2} - \dots + a_t \quad (3.15)$$

Sin embargo, si el proceso realmente fuera MA(1), no obtendríamos una representación con buenas características utilizando un modelo autorregresivo. Por el contrario, un proceso AR(1) no podría representarse con moderación utilizando un modelo de promedio móvil. En la práctica, para obtener una parametrización moderada, suele ser útil el modelo con términos autorregresivos como de promedio móvil. El modelo resultante

$$\tilde{z}_t = \phi_1 \tilde{z}_{t-1} + \dots + \phi_p \tilde{z}_{t-p} + a_t - \theta_1 a_{t-1} - \dots - \theta_q a_{t-q} \quad (3.16)$$

o de forma reducida

$$\phi(B) \tilde{z}_t = \theta(B) a_t \quad (3.17)$$

Se denomina proceso de orden mixto autorregresivo-media móvil(p,q), que abreviamos como ARMA(p,q). Por ejemplo, el proceso ARMA(1,1) es

$$\tilde{z}_t = \phi_1 \tilde{z}_{t-1} + a_t - \theta_1 a_{t-1} \quad (3.18)$$

Ahora los escribimos como

$$\begin{aligned} \tilde{z}_t &= \phi^{-1}(B) \theta(B) a_t \\ &= \frac{\theta(B)}{\phi(B)} a_t = \frac{1 - \theta_1 B - \dots - \theta_q B^q}{1 - \phi_1 B - \dots - \phi_p B^p} a_t \end{aligned} \quad (3.19)$$

vemos que el proceso ARMA mixto se puede considerar como la salida \tilde{Z}_t de un filtro lineal, cuya función de transferencia es el cociente de dos operadores polinómicos $\theta(B)$ y $\phi(B)$, cuando la entrada es ruido blanco a_t . Además, dado que $\tilde{z}_t = z_t - \mu$ de dónde $\mu = E[z_t]$ es la media del proceso del caso estacionario, el proceso general ARMA(p,q) también se puede escribir en términos del proceso original \tilde{z}_t como

$$\phi(B) z_t = \theta_0 + \theta(B) a_t \quad (3.20)$$

dónde la constante del término θ_0 está definida por

$$\theta_0 = (1 - \phi_1 - \phi_2 - \dots - \phi_p) \mu \quad (3.21)$$

3.2. Modelo lineal para series de tiempo no estacionarias

Muchas series de tiempo en aplicaciones reales se comportan como si no tuvieran una media fija. Aun así, exhiben homogeneidad en el sentido de que, aparte del nivel local, o quizás del nivel local y la tendencia, una parte de la serie se comporta de manera muy similar a cualquier otra parte. Los modelos que describen este comportamiento no estacionario homogéneo se pueden obtener suponiendo que alguna diferencia adecuada del proceso es estacionaria. En esta sección, examinamos las propiedades de la importante clase de modelos para los cuales la d -ésima diferencia de la serie es un proceso autorregresivo mixto estacionario-promedio móvil. Estos modelos se denominan procesos de promedio móvil integrado autorregresivo (ARIMA).

3.2.1. Método Autorregresivo integrado de media móvil

Aunque los modelos no estacionarios del tipo descrito anteriormente son valiosos para representar un comportamiento explosivo o evolutivo (como el crecimiento bacteriano), las aplicaciones que describimos no son de este tipo. Hemos visto que un proceso ARMA es estacionario si las raíces de $\phi(B) = 0$ se encuentran fuera del círculo unitario y exhibe un comportamiento no estacionario explosivo si las raíces se encuentran dentro del círculo unitario. El único caso que queda es que las raíces de $\phi(B) = 0$ se encuentran en el círculo unitario. Resulta que los modelos resultantes son de gran valor para representar series temporales homogéneas no estacionarias. En particular, las series no estacionales suelen estar bien representadas por modelos en los que una o más de estas raíces son la unidad. Consideremos el modelo

$$\phi(B)\tilde{z}_t = \theta(B)a_t \quad (3.22)$$

donde $\phi(B)$ es un operador autorregresivo no estacionario tal que d de las raíces de $\phi(B) = 0$ son la unidad y el resto está fuera del círculo unitario. Entonces el modelo se puede escribir como

$$\phi(B)\tilde{z}_t = \phi(B)(1 - B)^d\tilde{z}_t = \theta(B)a_t \quad (3.23)$$

donde $\phi(B)$ es un operador autorregresivo estacionario. Como $\nabla^d\tilde{z}_t = \nabla^dz_t$, para $d \geq 1$, donde $\nabla = 1 - B$ es el operador diferenciador, podemos escribir el modelo como:

$$\phi(B)\nabla^dz_t = \theta(B)a_t \quad (3.24)$$

Equivalentemente, el proceso queda definido por dos ecuaciones

$$\phi(B)w_t = \theta(B)a_t \quad (3.25)$$

y

$$w_t = \nabla^dz_t \quad (3.26)$$

Por lo tanto, vemos que el modelo corresponde a suponer que la d -ésima diferencia de la serie puede representarse mediante un proceso ARMA invertible estacionario. Una forma alternativa de ver el proceso para $d \geq 1$ resulta de invertir (3.26) para dar

$$z_t = S^d w_t \quad (3.27)$$

Dónde S es el operador de aditivo infinito definido por:

$$\begin{aligned} Sx_t &= \sum_{h=-\infty}^t x_h = (1 + B + B^2 + \dots)x_t \\ &= (1 - B)^{-1}x_t = \nabla^{-1}x_t \end{aligned} \quad (3.28)$$

De este modo

$$S = (1 - B)^{-1} = \nabla^{-1} \quad (3.29)$$

El operador S^2 se define de manera similar como

$$\begin{aligned} S^2x_t &= Sx_t + Sx_{t-1} + Sx_{t-2} + \dots \\ &= \sum_{t=-\infty}^t \sum_{h=-\infty}^i x_h = (1 + 2B + 3B^2 + \dots)x_t \end{aligned} \quad (3.30)$$

y así sucesivamente para d de orden superior. La ecuación (3.27) implica que el proceso (3.24) puede obtenerse sumando (o “integrando”) el proceso estacionario (3.25) d veces. Por lo tanto, llamamos al proceso (3.24) un proceso de promedio móvil integrado autorregresivo (ARIMA).

Los modelos ARIMA para series de tiempo no estacionarias, que también fueron considerados anteriormente por Yaglom (1955), son de fundamental importancia para el pronóstico y el control, como lo discutieron Box y Jenkins (1962, 1963, 1965, 1968a, 1968b, 1969) y Box et al. (1967a). Los procesos no estacionarios también fueron analizados por Zadeh y Ragazzini (1950), Kalman (1960) y Kalman y Bucy (1961). Un procedimiento anterior para el análisis de series de tiempo que empleaba la diferenciación fue el método de diferencia variable (ver Tintner (1940) y Rao y Tintner (1963)). Sin embargo, la motivación, los métodos y los objetivos de este procedimiento eran bastante diferentes de los discutidos aquí. Técnicamente, el operador de suma infinita $S = (1 - B)^{-1}$ en (3.27) no se puede usar para definir los procesos ARIMA no estacionarios, ya que las sumas infinitas involucradas no serán convergentes. En su lugar, podemos considerar el operador de sumatoria finita S_m para cualquier entero positivo m , dado por

$$S_m = (1 + B + B^2 + \dots + B^{m-1}) = \frac{1 - B^m}{1 - B} \quad (3.31)$$

De manera similar, el operador de suma doble finita se puede definir como

$$\begin{aligned} S_m^{(2)} &= \sum_{j=0}^{m-1} \sum_{i=j}^{m-1} B^i = (1 + 2B + 3B^2 + \cdots + mB^{m-1}) \\ &= \frac{1 - B^m - mB^m(1 - B)}{(1 - B)^2} \end{aligned} \quad (3.32)$$

ya que $(1 - B)S_m^{(2)} = S_m - mB^m$, y así sucesivamente. Entonces, la relación entre un proceso ARMA integrado z_t con $d = 1$, por ejemplo, y el proceso ARMA estacionario correspondiente $w_t = (1 - B)z_t$, en términos de valores que se remontan a un origen temporal anterior $k < t$, se puede expresar como

$$z_t = \frac{S_{t-k}}{1 - B^{t-k}} w_t = \frac{1}{1 - B^{t-k}} (w_t + w_{t-1} + \cdots + w_{k+1}) \quad (3.33)$$

de modo que $z_t = w_t + w_{t-1} + \cdots + w_{k+1} + z_k$ se puede considerar como la suma de un número finito de términos del proceso estacionario w más un valor de inicialización del proceso z en el tiempo k . Por lo tanto, en la definición formal de las propiedades estocásticas de un proceso ARIMA no estacionario generado en (3.22), normalmente sería necesario especificar las condiciones de inicialización del proceso en algún momento k en el pasado finito (pero posiblemente remoto). Sin embargo, estas especificaciones de condiciones iniciales tendrán poco efecto en la mayoría de las características importantes del proceso.

El modelo (3.24) es equivalente a representar el proceso z_t como la salida de un filtro lineal (a menos que $d = 0$, este es un filtro lineal inestable), cuya entrada es ruido blanco a_t . Alternativamente, podemos considerarlo como un dispositivo para transformar el proceso altamente dependiente y posiblemente no estacionario z_t , en una secuencia de variables aleatorias no correlacionadas a_t , es decir, para transformar el proceso en ruido blanco. Si en (3.24), el operador autorregresivo $\phi(B)$ es de orden p , se toma la d -ésima diferencia, y el operador de media móvil $\theta(B)$ es de orden q , decimos que tenemos un modelo ARIMA de orden (p, d, q) , o simplemente un proceso ARIMA(p, d, q).

3.3. Construcción del modelo múltiple ARIMA

Los modelos ARIMA son efectivos para moldear una serie de tiempo de un conjunto de datos. Sin embargo, hay casos en los que el modelo no logra obtener buenos resultados de precisión y esto se debe a que la serie de tiempo tiene pocos datos o las características son insuficientes para obtener un modelo correcto de predicción. Por ello, la propuesta es la creación de un modelo ARIMA múltiple mediante distintos conjuntos de datos con la unión de una red neuronal para combinar las características de cada conjunto.

A continuación, se detalla el diseño que conlleva un modelo ARIMA para después generalizar esta técnica y lograr el cálculo para los múltiples modelos.

3.3.1. Diseño de la estructura del modelo ARIMA

El pronóstico de series de tiempo consiste en predecir con cierto grado de precisión los valores futuros en función de los valores pasados de una serie de tiempo. Entonces, el pronóstico de series de tiempo es un proceso de modelado de series de tiempo. El objeto del modelado de series de tiempo con el modelo ARIMA es encontrar un modelo ARIMA adecuado, tal que el error de modelado definido por

$$e_t = y_t - \hat{y}_t$$

se minimiza, aquí $\hat{y}_t = ARIMA_\theta(p, q, d)$ es la salida del modelo ARIMA. El modelado de series de tiempo incluye las siguientes dos partes:

1. Determinación de la estructura.

Para encontrar buenos órdenes de modelo p , q y d , de modo que el modelo ARIMA de orden (p, q, d) tenga la mejor precisión de pronóstico con parámetros fijos θ ,

$$(p, q, d) = \min_{p, q, d} \{ [ARIMA_\theta(p, q, d) - y_t]^2 \} \quad (3.34)$$

2. Estimación de parámetros.

Estimar los parámetros del modelo θ , de modo que el ARIMA tenga la mejor precisión de pronóstico con estructura fija (p, q, d)

$$\theta = \min_{\theta} \{ [ARIMA_\theta(p, q, d) - y_t]^2 \} \quad (3.35)$$

Es imposible obtener el modelo $ARIMA_{\theta^*}(p^*, q^*, d^*)$, porque la determinación de la estructura (3.34) necesita el parámetro óptimo θ^* , mientras que la estimación del parámetro (3.35) necesita la estructura óptima (p^*, q^*, d^*) . Tradicionalmente, se adopta el método de caminata aleatoria para encontrar la estructura ARIMA (p, q, d) , luego, en función de la estructura obtenida, los parámetros θ se estiman utilizando los datos de entrenamiento. Si no se satisface la precisión del pronóstico, se debe volver a aplicar la caminata aleatoria para encontrar otra combinación (p, q, d) . Este proceso continúa hasta que el error de pronóstico es lo suficientemente pequeño. Este es un proceso lento e ineficiente. En general no podemos encontrar la estructura ARIMA óptima. Utilizamos los siguientes dos criterios para determinar (p, q, d) : criterio de información de Akaike (AIC) y criterio de información bayesiano (BIC). Ambos siguen el mismo concepto: evaluar tanto el error de entrenamiento como la complejidad. El objetivo de estos criterios es que el modelo sea lo más simple posible, y a su vez garantice un buen ajuste con los datos de entrenamiento. Suponga que el conjunto de datos D consta de N datos, $\{y_1, \dots, y_N\}$.

$$\begin{aligned} AIC &: 2(p + q + 1) - 2\mathcal{L} \\ BIC &: \ln(N)(p + q + 1) - 2\mathcal{L} \end{aligned} \quad (3.36)$$

donde $(p + q + 1)$ es el número de parámetros de estructura de ARIMA (p, q, d) , \mathcal{L} es el valor máximo de la función de verosimilitud para el modelo ARIMA, se estima como

$$\mathcal{L} = N \ln \left(\frac{1}{N} \sum_{t=1}^N e_t^2 \right), \quad e_t = \hat{y}_t - y_t$$

donde \hat{y}_t es la salida del modelo ARIMA. Podemos ver en (3.36), AIC penaliza a los modelos que utilizan más parámetros de estructura, se desean puntuaciones AIC más bajas. Entonces, si hay dos modelos en la misma serie temporal, se preferirá el que tenga menos valor $p + q$ porque tiene una puntuación AIC más baja.

3.3.2. Identificación del coeficiente del modelo ARIMA

Una vez que se determinan (p, q, d) , se utiliza el método de mínimos cuadrados para estimar el parámetro θ . Para los i -ésimos datos

$$\theta(i) = [a_1(i) \cdots a_p(i), b_1(i) \cdots b_q(i)], \quad i = 1 \cdots N \quad (3.37)$$

Dónde n es el tamaño de los datos, la forma vectorial del modelo ARIMA está representado por la siguiente forma

$$y_t = \begin{bmatrix} a_1 & \cdots & a_p & b_1 & \cdots & b_q \end{bmatrix} \begin{bmatrix} x_{t-1} \\ \vdots \\ x_{t-p} \\ \epsilon_t \\ \vdots \\ \epsilon_{t-q} \end{bmatrix} \quad (3.38)$$

Como los ruidos ϵ_{t-k} , $k = 0 \cdots q$, no están disponibles, puede estimarlos usando los valores anteriores

$$\hat{\epsilon}_t = \hat{y}_t - \left(\sum_{k=1}^p a_k z^{-k} \right) y_t - \left(\sum_{k=1}^q b_k z^{-k} \right) \hat{\epsilon}_t \quad (3.39)$$

Entonces el modelo ARIMA se convierte en

$$Y = \theta \bar{Y} + E \quad (3.40)$$

Dónde

$$Y = [y_{t-1} \cdots y_{t-p}, \hat{\epsilon}_t \cdots \hat{\epsilon}_{t-q}]^T, \quad \bar{Y} = \begin{bmatrix} \hat{y}_{1,1} & \cdots & \hat{y}_{1,N} \\ \vdots & & \vdots \\ \hat{y}_{p,1} & & \hat{y}_{p,N} \\ \hat{\epsilon}_{1,1} & & \hat{\epsilon}_{1,N} \\ \vdots & & \vdots \\ \hat{\epsilon}_{q,1} & \cdots & \hat{\epsilon}_{q,N} \end{bmatrix}$$

$E = [e_1 \ \cdots \ e_{p+q}]$ es el vector de error residual. El objetivo de la identificación del parámetro es:

$$\min_{\theta} \|Y_t - \theta \bar{Y}\|^2 \quad (3.41)$$

Como (3.40) es un proceso lineal en parámetros, el valor óptimo solución de θ es

$$\theta = [\bar{Y}\bar{Y}^T]^{-1} \bar{Y}^T Y \quad (3.42)$$

Finalmente, obtenemos l modelos ARIMA óptimos, $M_1 \cdots M_l$, con l conjuntos de datos de entrenamiento, $M_1 \cdots M_l$. El proceso del modelo ARIMA para el pronóstico de series de tiempo se muestra en la Figura 3.1.

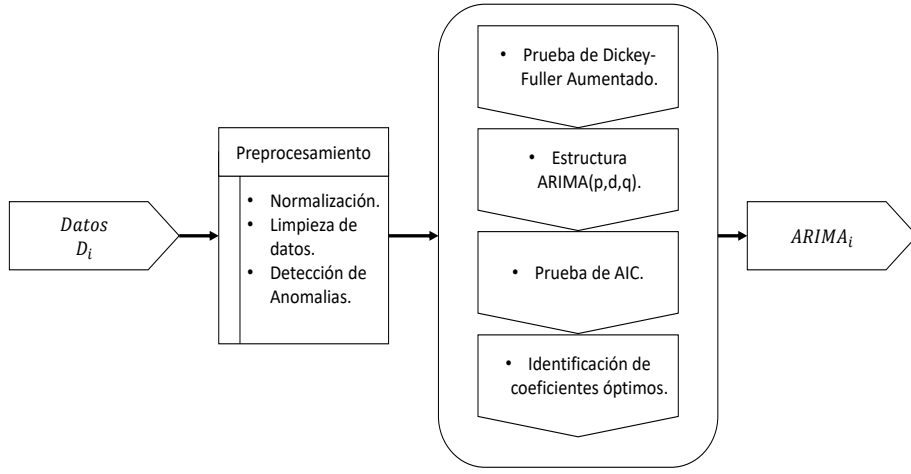


Figura 3.1: Arquitectura del modelo ARIMA para pronóstico de series de tiempo.

3.3.3. ARIMA múltiple

Cuando se tienen l conjuntos de diferentes datos

$$\{y_t^\sigma\}, \quad \sigma = \{1, 2, \dots, l\}$$

podemos entrenar l modelos ARIMA $M_1 \cdots M_l$ usando el enfoque de modelado ARIMA único propuesto en la sección anterior para determinar (p, q, d) para cada conjunto de datos σ , y para estimar los parámetros de cada (p, q, d) -orden el modelo ARIMA usados en (3.42). El vector de parámetros θ_i está asociado a un modelo ARIMA M_i . Los múltiples ARIMA se presentan como:

$$ARIMA_{\theta^i}(p^i, q^i, d^i), \quad i \in \sigma \quad (3.43)$$

El modelo $ARIMA_{\theta^i}(p^i, q^i, d^i)$ puede predecir los conjuntos de datos $\{y_t^i\}$.

Capítulo 4

Modelo de aprendizaje automático para predicción

La gran mayoría de los métodos de aprendizaje automático utilizados en la práctica implican técnicas de aprendizaje supervisado. En el aprendizaje supervisado, se usa un algoritmo para aprender una función de mapeo aproximada de la variable de entrada x a la salida $y = f(x)$ para cuando haya nuevos datos de entrada, podamos predecir la salida de esos datos. En el caso de las redes neuronales no es el mejor método para modelar una serie de tiempo, pero el modelo que se busca requiere de un mecanismo que extraiga las mejores características de los modelos ARIMA pre-entrenados y los combine para obtener un modelo de buena predicción. En este capítulo se verá a detalle la representación de una red neuronal y las métricas de estimación que suelen medir el error de las predicciones.

4.1. Redes neuronales

Las redes neuronales artificiales (ANN) son modelos de clasificación que pueden aprender límites de decisión altamente complejos y no lineales únicamente a partir de los datos. Han obtenido una amplia aceptación en varias aplicaciones, como el procesamiento de la visión, asistentes de voz y el lenguaje, donde se ha demostrado repetidamente que superan a otros modelos de clasificación (y, en algunos casos, incluso al rendimiento humano). Históricamente, el estudio de las redes neuronales artificiales se inspiró en los intentos de emular los sistemas neuronales biológicos. El cerebro humano se compone principalmente de células nerviosas llamadas neuronas, unidas entre sí con otras neuronas a través de hilos de fibra llamados axones. Cada vez que se estimula una neurona (p. ej., en respuesta a un estímulo), transmite activaciones nerviosas a través de axones a otras neuronas. Las neuronas receptoras recogen estas activaciones nerviosas utilizando estructuras llamadas dendritas, que son extensiones del cuerpo celular de la neurona. La fuerza del punto de contacto entre una dendrita y un axón, conocida como sinapsis, determina la conectividad entre las neuronas. Los neurocientíficos han descubierto que el cerebro humano aprende cambiando la fuerza

de la conexión sináptica entre las neuronas tras la estimulación repetida por el mismo impulso [61].

El cerebro humano consta de aproximadamente 100 mil millones de neuronas que están interconectadas de manera compleja, lo que nos permite aprender nuevas tareas y realizar actividades regulares. Tenga en cuenta que una sola neurona solo realiza una función modular simple, que es responder a las activaciones nerviosas provenientes de las neuronas emisoras conectadas en su dendrita y transmitir su activación a las neuronas receptoras a través de los axones. Sin embargo, es la composición de estas funciones simples las que juntas pueden expresar funciones complejas. Esta idea está en la base de la construcción de redes neuronales artificiales [61].

Análoga a la estructura de un cerebro humano, una red neuronal artificial se compone de una serie de unidades de procesamiento, llamadas nodos, que están conectados entre sí a través de enlaces dirigidos. Los nodos corresponden a neuronas que realizan las unidades básicas de cómputo, mientras que los enlaces dirigidos corresponden a conexiones entre neuronas, constituidos por axones y dendritas. Además, el peso de un enlace dirigido entre dos neuronas representa la fuerza de la conexión sináptica entre neuronas. Al igual que en los sistemas neuronales biológicos, el objetivo principal de ANN es adaptar los pesos de los enlaces hasta que se ajusten a las relaciones de entrada y salida de los datos subyacentes [61].

La motivación básica detrás del uso de un modelo ANN es extraer características útiles de los atributos originales que son más relevantes para la clasificación. Mediante el uso de una combinación compleja de nodos interconectados, los modelos ANN pueden extraer conjuntos de características mucho más sustanciosos, lo que da como resultado un buen rendimiento de clasificación. Además, los modelos ANN proporcionan una forma natural de representar características en múltiples niveles de abstracción, donde las características complejas se ven como composiciones de características más simples. En muchos problemas de clasificación, modelar tal jerarquía de características resulta muy útil. Por ejemplo, para detectar un rostro humano en una imagen, primero podemos identificar características de bajo nivel, como bordes afilados con diferentes degradados y orientaciones. Estas características se pueden combinar para identificar partes faciales como ojos, nariz, orejas y labios. Finalmente, se puede utilizar una disposición adecuada de las partes faciales para identificar correctamente un rostro humano. Los modelos ANN proporcionan una arquitectura poderosa para representar una abstracción jerárquica de características, desde niveles inferiores de abstracción (p. ej., bordes) hasta niveles superiores (p. ej., partes faciales) [61].

Las redes neuronales artificiales han tenido una larga historia de desarrollos que abarca más de cinco décadas de investigación. Aunque los modelos clásicos de ANN sufrieron varios desafíos que obstaculizaron el progreso durante mucho tiempo, han

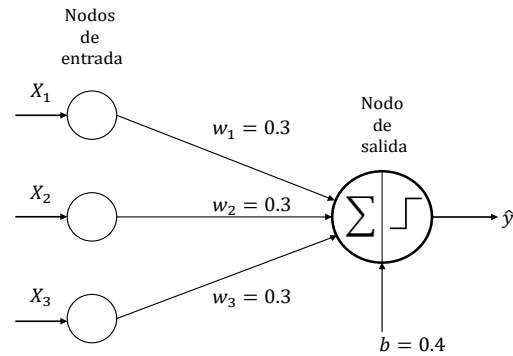
resurgido con una gran popularidad debido a una serie de desarrollos recientes en la última década, conocidos colectivamente como aprendizaje profundo. En esta sección, examinamos los enfoques clásicos para aprender modelos ANN, desde el modelo más simple llamado perceptrones hasta arquitecturas más complejas llamadas redes neuronales multicapa [61].

4.1.1. Perceptrón

Un perceptrón es un tipo básico de modelo ANN que involucra dos tipos de nodos: nodos de entrada, que se usan para representar los atributos de entrada, y un nodo de salida, que se usa para representar la salida del modelo. La figura 4.1b ilustra la arquitectura básica de un perceptrón que toma tres atributos de entrada $\{x_1, x_2, x_3\}$ y produce una salida binaria y .

x_1	x_2	x_3	y
1	0	0	-1
1	0	1	1
1	1	0	1
1	1	1	1
0	0	1	-1
0	1	0	-1
0	1	1	1
0	0	0	-1

(a) Datos de entrada.



(b) Perceptrón.

Figura 4.1: Función booleana modelada mediante un perceptrón.

El perceptrón consta de dos tipos de nodos: nodos de entrada, que se utilizan para representar los atributos de entrada, y un nodo de salida, que se utiliza para representar la salida del modelo. Los nodos en una arquitectura de red neuronal se conocen comúnmente como neuronas o unidades. En un perceptrón, cada nodo de entrada está conectado a través de un enlace ponderado al nodo de salida. El enlace ponderado se utiliza para emular la fuerza de la conexión sináptica entre las neuronas. Al igual que en los sistemas neuronales biológicos, entrenar un modelo de perceptrón equivale a adaptar los pesos de los enlaces hasta que se ajusten a las relaciones de entrada y salida de los datos subyacentes. Un perceptrón calcula su valor de salida (\hat{y}), realizando una suma ponderada en sus entradas, restando un factor de sesgo t de la suma y luego examinando el signo del resultado. El modelo que se muestra en la figura 4.1b tiene tres nodos de entrada, cada uno de los cuales tiene un peso idéntico de $w = 0.3$ al nodo de salida y un factor de sesgo de $t = 0.4$. La salida calculada por

el modelo es

$$\hat{y} = \begin{cases} 1, & \text{si } 0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 > 0; \\ -1, & \text{si } 0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 < 0. \end{cases} \quad (4.1)$$

Por ejemplo, si $x_1 = 1$, $x_2 = 1$, $x_3 = 0$ el resultado de la salida daría $\hat{y} = 1$ dado que $0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4$ es positivo. Por otro lado, si $x_1 = 0$, $x_2 = 1$, $x_3 = 0$ entonces $\hat{y} = -1$ porque la suma ponderada restada por el factor de sesgo es negativa. Se tiene que tomar en cuenta la diferencia entre los nodos de entrada y salida de un perceptrón. Un nodo de entrada simplemente transmite el valor que recibe al enlace de salida sin realizar ninguna transformación. El nodo de salida, por otro lado, es un dispositivo matemático que calcula la suma ponderada de sus entradas, resta el término de sesgo y luego produce una salida que depende del signo de la suma resultante. Más específicamente, la salida de un modelo de perceptrón se puede expresar matemáticamente de la siguiente manera:

$$\hat{y} = \text{signo}(w_d x_d + w_{d-1} x_{d-1} + \dots + w_2 x_2 + w_1 x_1 - t) \quad (4.2)$$

donde w_1, w_2, \dots, w_d son los pesos de los enlaces de entrada, x_1, x_2, \dots, x_d son los valores de los atributos de entrada y d es la cantidad de datos. La función de signo, que actúa como una función de activación para la neurona de salida, emite un valor de $+1$ si su argumento es positivo y -1 si su argumento es negativo. El modelo de perceptrón se puede escribir en una forma más compacta de la siguiente manera:

$$\hat{y} = \text{signo}[w_d x_d + w_{d-1} x_{d-1} + \dots + w_1 x_1 + w_0 x_0] = \text{signo}(w \cdot x) \quad (4.3)$$

donde $w_0 = -t$, $x_0 = 1$, y $w \cdot x$ es el producto escalar entre el vector de peso w y el vector de atributo de entrada x [61].

Aprendizaje del modelo perceptrón

Durante la fase de entrenamiento de un modelo de perceptrón, los parámetros de peso w se ajustan hasta que las salidas del perceptrón se vuelven consistentes con las verdaderas salidas de los ejemplos de entrenamiento. En el algoritmo 1 se proporciona un resumen del algoritmo de aprendizaje del perceptrón. El cálculo clave para este algoritmo es la fórmula de actualización de peso dada en la línea 7 del pseudocódigo:

$$w_j^{k+1} = w_j^k + \lambda(y_i - \hat{y}_i^k)x_{ij} \quad (4.4)$$

donde $w(k)$ es el parámetro de peso asociado con el i -ésimo enlace de entrada después de la k -ésima iteración, λ es un parámetro conocido como tasa de aprendizaje y x_{ij} es el valor del j -ésimo atributo del ejemplo de entrenamiento x_i . La justificación de la fórmula de actualización de peso es bastante intuitiva. La ecuación 4.4 muestra que la nueva ponderación w^{k+1} es una combinación de la anterior ponderación $w^{(k)}$ y un término proporcional al error de predicción, $(y - \hat{y})$. Si la predicción es correcta, entonces el peso permanece sin cambios. En caso contrario, se modifica de las siguientes formas:

- Si $y = +1$ y $\hat{y} = -1$, entonces el error de predicción es $(y - \hat{y}) = 2$. Para compensar el error, necesitamos aumentar el valor de la salida pronosticada aumentando los pesos de todos los enlaces con entradas positivas y disminuyendo los pesos de todos los enlaces con entradas negativas.
- Si $y_i = -1$ y $\hat{y} = +1$, entonces el error $(y - \hat{y}) = -2$. Para compensar, necesitamos disminuir el valor de la salida prevista al disminuir los pesos de todos los enlaces con entradas positivas y aumentar los pesos de todos los enlaces con entradas negativas.

Algorithm 1 Aprendizaje de un perceptrón

```

1: Tenemos que  $D = \{(x_i, y_i) \mid i = 1, 2, \dots, N\}$  un conjunto de datos de entrena-
   miento
2: Inicializamos el vector de pesos con valores random,  $w^{(0)}$ 
3: repeat
4:   for cada ejemplo de entrenamiento  $(x_i, y_i) \in D$  do
5:     Calcular la salida predictiva  $\hat{y}_i^{(k)}$ 
6:     for cada peso  $w_j$  do
7:       Actualizar el peso,  $w_j^{(k+1)} = w_j^{(k)} + \lambda(y_i - \hat{y}_i^{(k)})x_{ij}$ 
8:     end for
9:   end for
10: until se cumpla la condición de parada  $=0$ 

```

En la fórmula de actualización de peso, los enlaces que más contribuyen al término de error son los que requieren el mayor ajuste. Sin embargo, los pesos no deben cambiarse demasiado drásticamente porque el término de error se calcula solo para el ejemplo de entrenamiento actual. De lo contrario, se desharán los ajustes realizados en iteraciones anteriores. La tasa de aprendizaje λ , un parámetro cuyo valor está entre 0 y 1, se puede utilizar para controlar la cantidad de ajustes realizados en cada iteración. Si λ está cerca de 0, entonces el nuevo peso está influenciado principalmente por el valor del peso anterior. Por otro lado, si λ está cerca de 1, entonces el nuevo peso es sensible a la cantidad de ajuste realizado en la iteración actual. En algunos casos, se puede utilizar un valor de λ adaptativo; inicialmente, λ es moderadamente grande durante las primeras iteraciones y luego disminuye gradualmente en las iteraciones posteriores [61].

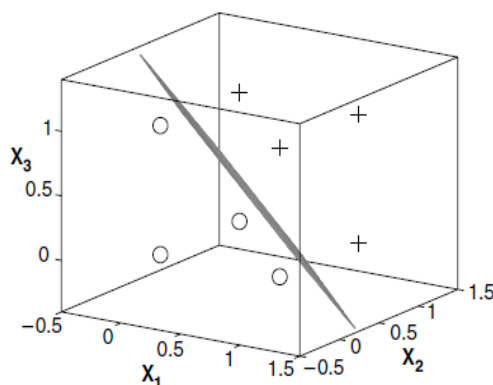


Figura 4.2: Frontera de decisión del perceptrón.

El modelo de perceptrón que se muestra en la ecuación 4.3 es lineal en sus parámetros w y atributos x . Debido a esto, el límite de decisión de un perceptrón, que se obtiene al establecer $\hat{y} = 0$, es un hiperplano lineal que separa los datos en dos clases, -1 y +1. La figura 4.2 muestra el límite de decisión obtenido al aplicar el algoritmo de aprendizaje del perceptrón al conjunto de datos proporcionado en la figura 4.1. Se garantiza que el algoritmo de aprendizaje del perceptrón convergerá a una solución óptima (siempre que la tasa de aprendizaje sea lo suficientemente pequeña) para problemas de clasificación linealmente separables. Si el problema no es linealmente separable, el algoritmo no converge. La figura 4.3 muestra un ejemplo de datos separables no lineales proporcionados por la función XOR. El perceptrón no puede encontrar la solución correcta para estos datos porque no existe un hiperplano lineal que pueda separar perfectamente las instancias de entrenamiento [61].

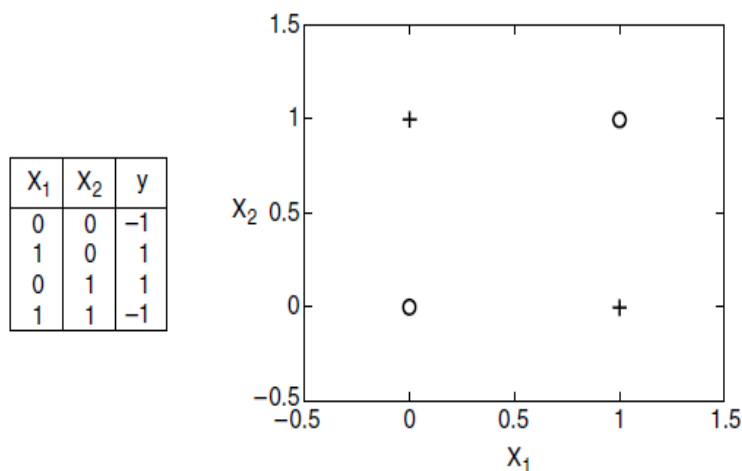


Figura 4.3: Problema de clasificación de una compuerta XOR.

4.1.2. Red neuronal artificial multicapa

Una red neuronal artificial tiene una estructura más compleja que la de un modelo de perceptrón. Las complejidades adicionales pueden surgir de varias maneras:

1. La red puede contener varias capas intermedias entre sus capas de entrada y salida. Estas capas intermedias se denominan capas ocultas y los nodos integrados en estas capas se denominan nodos ocultos. La estructura resultante se conoce como red neuronal multicapa (consulte la figura 4.4). En una red neuronal feed-forward, los nodos en una capa están conectados solo a los nodos en la siguiente capa. El perceptrón es una red neuronal de alimentación directa de una sola capa porque tiene solo una capa de nodos, la capa de salida, que realiza operaciones matemáticas complejas. En una red neuronal recurrente, los enlaces pueden conectar nodos dentro de la misma capa o nodos de una capa a las capas anteriores [61].

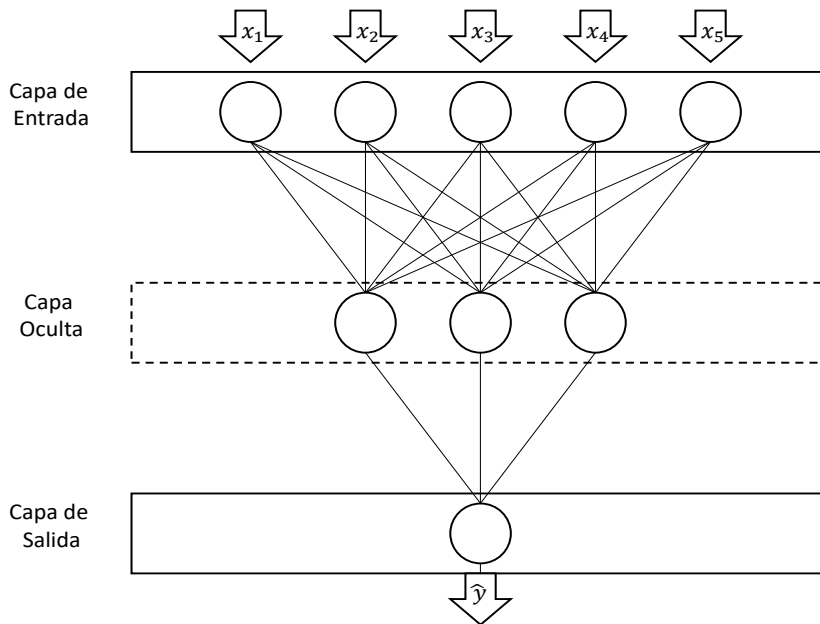


Figura 4.4: Ejemplo de una red neuronal multicapa.

2. La red puede utilizar tipos de funciones de activación distintos de la función de señal. Los ejemplos de otras funciones de activación incluyen las funciones de tangente lineal, sigmoide (logística) e hiperbólica, como se muestra en la figura 4.5. Estas funciones de activación permiten que los nodos ocultos y de salida produzcan valores de salida que no son lineales en sus parámetros de entrada.

Estas complejidades adicionales permiten que las redes neuronales multicapa modelen relaciones más complejas entre las variables de entrada y salida. Por ejemplo, considere

el problema XOR descrito en la sección anterior. Las instancias se pueden clasificar mediante dos hiperplanos que dividen el espacio de entrada en sus respectivas clases, como se muestra en la figura 4.6a. Debido a que un perceptrón puede crear solo un hiperplano, no puede encontrar la solución óptima. Este problema se puede abordar utilizando una red neuronal de alimentación hacia adelante de dos capas, como se muestra en la figura 4.6b. Intuitivamente, podemos pensar en cada nodo oculto como un perceptrón que intenta construir uno de los dos hiperplanos, mientras que el nodo de salida simplemente combina los resultados de los perceptrones para producir el límite de decisión que se muestra en la figura 4.6a.

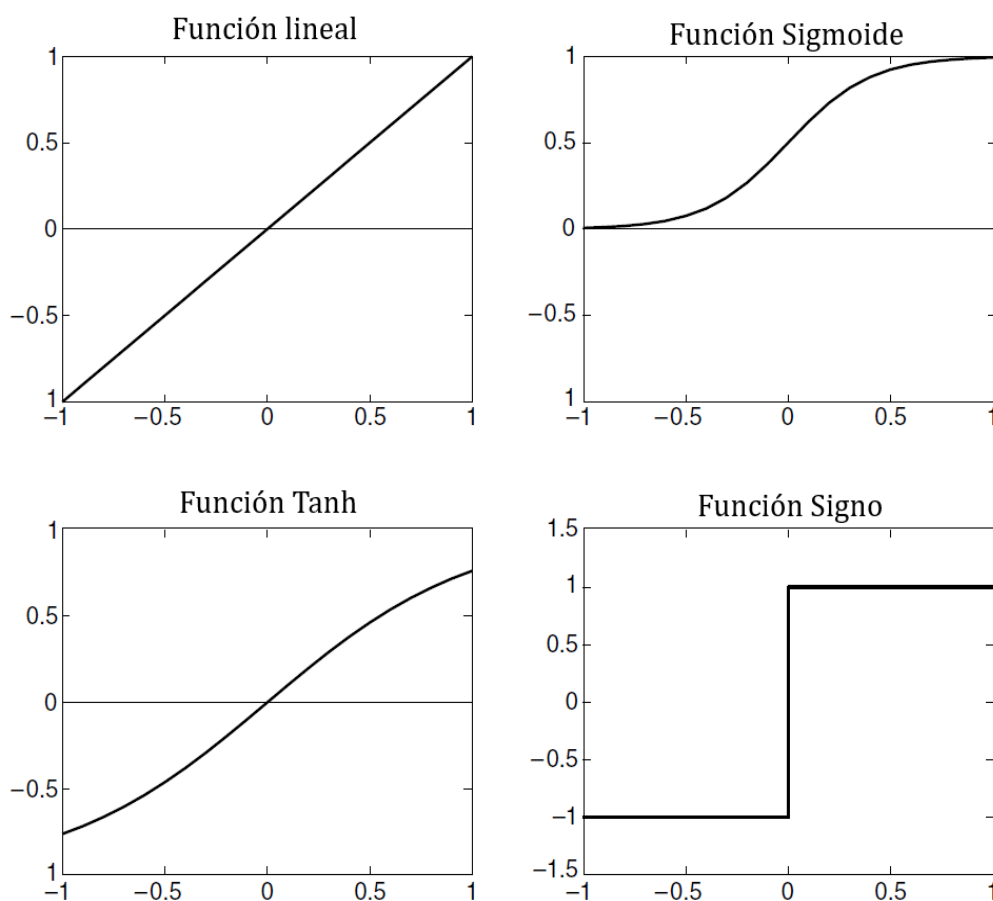


Figura 4.5: Funciones de activación para redes neuronales.

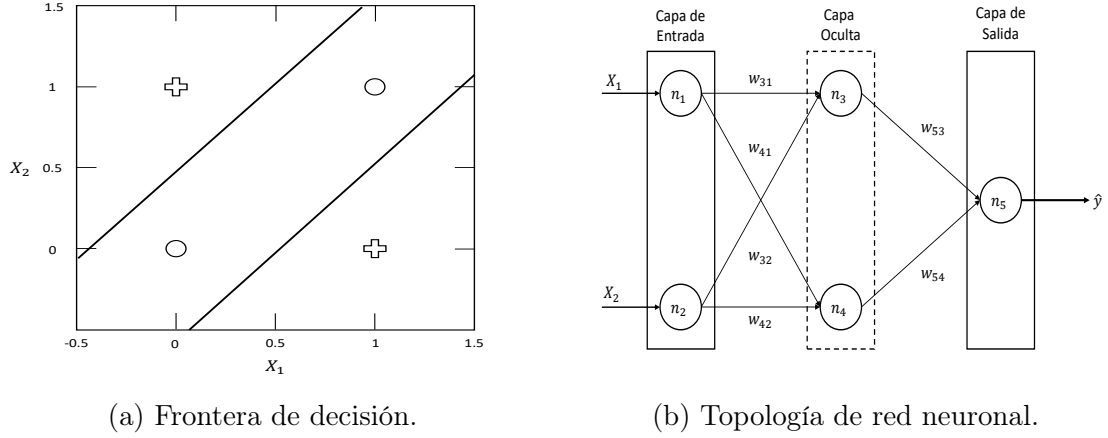


Figura 4.6: Problema XOR visto desde una red neuronal de dos capas.

Para aprender los pesos de un modelo ANN, necesitamos un algoritmo eficiente que converja a la solución correcta cuando se proporciona una cantidad suficiente de datos de entrenamiento. Un enfoque es tratar cada nodo oculto o nodo de salida en la red como una unidad de perceptrón independiente y aplicar la misma fórmula de actualización de peso que la ecuación 4.4. Obviamente, este enfoque no funcionará porque carecemos de conocimiento a priori sobre los verdaderos resultados de los nodos ocultos. Esto dificulta determinar el término de error, $(y - \hat{y})$, asociado con cada nodo oculto. A continuación, se presenta una metodología para aprender los pesos de una red neuronal basada en el enfoque de descenso de gradiente [61].

Aprendizaje en una red neuronal multicapa

El objetivo del algoritmo de aprendizaje ANN es determinar un conjunto de pesos w que minimicen la suma total de errores al cuadrado:

$$E(w) = \frac{1}{2} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (4.5)$$

La suma de los errores al cuadrado depende de w porque la clase predicha \hat{y} es una función de los pesos asignados a los nodos ocultos y de salida. La figura 4.7 muestra un ejemplo de la superficie de error en función de sus dos parámetros, w_1 y w_2 . Este tipo de superficie de error suele encontrarse cuando \hat{y}_i es una función lineal de sus parámetros, w . Si reemplazamos $\hat{y} = w \cdot x$ en la ecuación 4.5, entonces la función de error se vuelve cuadrática en sus parámetros y se puede encontrar fácilmente una solución mínima global [61].

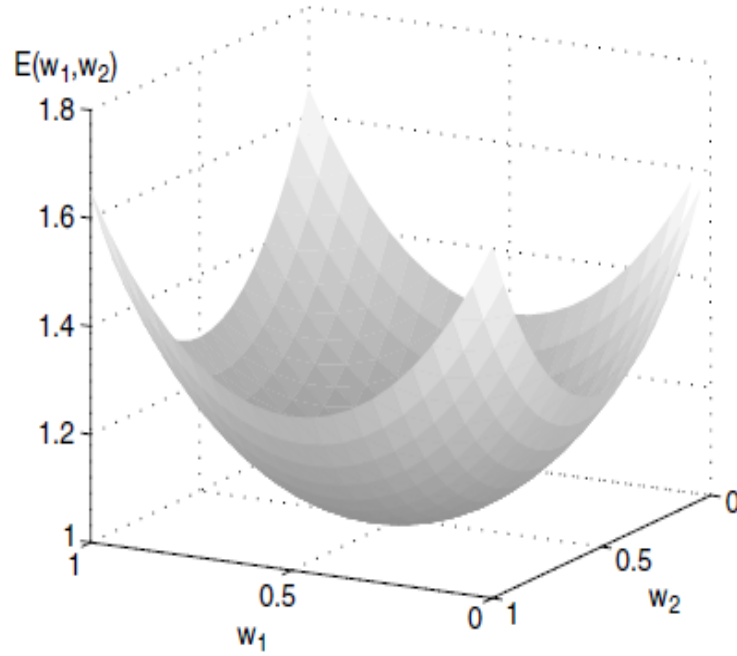


Figura 4.7: Superficie de error $E(w_1, w_2)$ para el modelo de 2 parámetros.

En la mayoría de los casos, la salida de una ANN es una función no lineal de sus parámetros debido a la elección de sus funciones de activación (p. ej., función sigmoidea o tanh). Como resultado, ya no es sencillo derivar una solución para w que garantice ser globalmente óptima. Se han desarrollado algoritmos codiciosos como los basados en el método de descenso de gradiente para resolver de manera eficiente el problema de optimización. La fórmula de actualización de peso utilizada por el método de descenso de gradiente se puede escribir de la siguiente manera:

$$w_j \leftarrow w_j - \lambda \frac{\partial E(w)}{\partial w_j} \quad (4.6)$$

donde λ es la tasa de aprendizaje. El segundo término establece que el peso debe aumentarse en una dirección que reduzca el término de error general. Sin embargo, debido a que la función de error no es lineal, es posible que el método de descenso de gradiente quede atrapado en un mínimo local. El método de descenso de gradiente se puede utilizar para aprender los pesos de la salida y los nodos ocultos de una red neuronal. Para los nodos ocultos, el cálculo no es trivial porque es difícil evaluar su término de error, $\partial E / \partial w_j$, sin saber cuáles deberían ser sus valores de salida. Se ha desarrollado una técnica conocida como retropropagación para abordar este problema. Hay dos fases en cada iteración del algoritmo: la fase de avance y la fase de retroceso. Durante la fase de avance, los pesos obtenidos de la iteración anterior se utilizan para calcular el valor de salida de cada neurona de la red. El cálculo avanza en la dirección de avance; es decir, las salidas de las neuronas en el nivel k se calculan antes de calcular las salidas en el nivel $k + 1$. Durante la fase hacia atrás, la fórmula

de actualización del peso se aplica en la dirección inversa. En otras palabras, los pesos en el nivel $k + 1$ se actualizan antes de que se actualicen los pesos en el nivel k . Este enfoque de retropropagación nos permite utilizar los errores de las neuronas en la capa $k + 1$ para estimar los errores de las neuronas en la capa k [61].

4.1.3. Consideraciones a tomar en el diseño de una red neuronal

Antes de entrenar una red neuronal para aprender una tarea de clasificación, se deben considerar los siguientes problemas de diseño.

1. Se debe determinar el número de nodos en la capa de entrada. Asigne un nodo de entrada a cada variable de entrada numérica o binaria. Si la variable de entrada es categórica, podríamos crear un nodo para cada valor categórico o codificar la variable k -aria usando nodos de entrada $\lceil \log_2 k \rceil$.
2. Se debe establecer el número de nodos en la capa de salida. Para un problema de dos clases, es suficiente usar un solo nodo de salida. Para un problema de clase k , hay k nodos de salida.
3. Se debe seleccionar la topología de la red (por ejemplo, el número de capas y nodos ocultos, y la arquitectura de red recurrente o de avance). Tenga en cuenta que la representación de la función de destino depende de los pesos de los enlaces, la cantidad de nodos y capas ocultos, los sesgos en los nodos y el tipo de función de activación. Encontrar la topología correcta no es una tarea fácil. Una forma de hacer esto es comenzar desde una red totalmente conectada con una cantidad suficientemente grande de nodos y capas ocultas, y luego repetir el procedimiento de creación de modelos con una cantidad menor de nodos. Este enfoque puede llevar mucho tiempo. Alternativamente, en lugar de repetir el procedimiento de creación de modelos, podríamos eliminar algunos de los nodos y repetir el procedimiento de evaluación del modelo para seleccionar la complejidad correcta.
4. Los pesos y sesgos deben inicializarse. Las asignaciones aleatorias suelen ser aceptables.
5. Los ejemplos de entrenamiento con valores faltantes deben eliminarse o reemplazarse con los valores más probables o interpolando los datos [61].

4.1.4. Características de una red neuronal

A continuación se resumen las características generales de una red neuronal artificial:

1. Las redes neuronales multicapa con al menos una capa oculta son aproximadores universales; es decir, se pueden usar para aproximar cualquier función

objetivo. Dado que una ANN tiene un espacio de hipótesis muy expresivo, es importante elegir la topología de red apropiada para un problema dado para evitar el sobreajuste del modelo.

2. ANN puede manejar funciones redundantes porque los pesos se aprenden automáticamente durante el paso de entrenamiento. Los pesos de las características redundantes tienden a ser muy pequeños.
3. Las redes neuronales son bastante sensibles a la presencia de ruido en los datos de entrenamiento. Un enfoque para manejar el ruido es usar un conjunto de validación para determinar el error de generalización del modelo. Otro enfoque es disminuir el peso por algún factor en cada iteración.
4. El método de descenso de gradiente utilizado para aprender los pesos de una ANN a menudo converge a algún mínimo local. Una forma de escapar del mínimo local es agregar un término de impulso a la fórmula de actualización del peso.
5. Entrenar una ANN es un proceso que requiere mucho tiempo, especialmente cuando la cantidad de nodos ocultos es grande. Sin embargo, los ejemplos de prueba se pueden clasificar rápidamente [61].

4.2. Métricas de estimación

Los métodos de clasificación y pronóstico idealmente debe ser evaluados en las situaciones para las cuales serán empleados. La mayoría de estos modelos están basados en procedimientos metodológicos o la obtención de una muestra de errores. Por lo anterior se debe efectuar un chequeo para verificar que la propuesta corresponde a una buena precisión. A continuación, se describen distintos tipos de métricas de precisión que miden error a la salida de cada modelo y muestran distintas características respecto a los datos evaluados [61].

Para describir cada elemento de las fórmulas de las mediciones estandarizamos que Y_i es el i -ésimo valor previsto, el elemento X_i es el i -ésimo valor actual de un conjunto de datos y N es el número de muestras que este contiene.

- MSE se puede utilizar si hay valores atípicos que deben detectarse. De hecho, MSE es excelente para atribuir pesos más grandes a tales puntos, gracias a la norma L_2 : claramente, si el modelo final genera una predicción muy mala, la parte cuadrada de la función resalta el error [62].

$$MSE = \frac{1}{N} \sum_{i=1}^N (Y_i - X_i)^2 \quad (4.7)$$

(el mejor valor = 0, el peor valor = $+\infty$)

- Las mediciones de MSE y RMSE están monótonamente relacionadas (a través de la raíz cuadrada). Un arreglo de modelos de regresión basada en MSE será idéntica a un arreglo de modelos basada en RMSE [62].

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N (Y_i - X_i)^2} \quad (4.8)$$

(el mejor valor = 0, el peor valor = $+\infty$)

- MAE se puede utilizar si los valores atípicos representan partes dañadas de los datos. Por defecto, MAE no penaliza demasiado los valores atípicos de entrenamiento (la norma L_1 de alguna manera suaviza todos los errores de los posibles valores atípicos), proporcionando así una medida de rendimiento genérica y acotada para el modelo. Por otro lado, si el conjunto de prueba también tiene muchos valores atípicos, el rendimiento del modelo será mediocre [62].

$$MAE = \frac{1}{N} \sum_{i=1}^N |Y_i - X_i| \quad (4.9)$$

(el mejor valor = 0, el peor valor = $+\infty$)

- MAPE es otra métrica de desempeño para modelos de regresión, teniendo una interpretación muy intuitiva en términos de error relativo: por su definición, se recomienda su uso en tareas donde es más importante ser sensible a variaciones relativas que a variaciones absolutas (De Myttenaere et al. 2016). Sin embargo, también tiene una serie de inconvenientes, siendo los más críticos la restricción de su uso a datos estrictamente positivos por definición y su sesgo hacia pronósticos bajos, lo que lo hace inadecuado para modelos predictivos donde se esperan grandes errores (Armstrong y Collopy, 1992) [62].

$$MAPE = \frac{1}{N} \sum_{i=1}^N \frac{|X_i - Y_i|}{X_i} \cdot 100\% \quad (4.10)$$

(el mejor valor = 0, el peor valor = $+\infty$)

- El coeficiente de determinación o R^2 (Wright, 1921) puede interpretarse como la proporción de la varianza de la variable dependiente que es predecible a partir de las variables independientes [62].

$$R^2 = 1 - \frac{\sum_{i=1}^N (Y_i - X_i)^2}{\sum_{i=1}^N (\bar{X}_i - X_i)^2} \quad (4.11)$$

(el mejor valor = 1, el peor valor = $-\infty$)

- SMAPE es el error porcentual absoluto medio simétrico definido inicialmente por Armstrong (1985), y luego refinado en su versión actual por Flores (1986) y Makridakis (1993), SMAPE fue propuesto para enmendar los inconvenientes de la métrica MAPE [62].

$$SMAPE = \frac{100\%}{N} \sum_{i=1}^N \frac{|Y_i - X_i|}{(|Y_i| + |X_i|)/2} \cdot 100\% \quad (4.12)$$

(el mejor valor = 0, el peor valor = 2)

Capítulo 5

Modelo híbridos

Los métodos híbridos son la combinación de uno o más modelos como los que muestra el esquema 2.1. Lo que buscan al crear estos modelos es combinar las características que ofrece cada uno de ellos y conformar un nuevo modelo que mejore la precisión o previsión de los datos. El inicio de este capítulo describe algunos de los modelos híbridos utilizados en la literatura, mientras que la segunda parte es la explicación del modelo planteado partiendo de los modelos múltiples ARIMA que se vieron en el capítulo 3.

5.1. Modelos híbridos

En la actualidad, existen distintos métodos aplicados a resolver la predicción de la energía eólica en el mundo. A continuación, se exponen los ejemplos más actuales que este campo ha podido resolver y los resultados que se han obtenido de cada uno de los trabajos.

El primero de ellos es una propuesta realizada en México por Cadenas et al. 2010 [20]. El modelo empleado en este trabajo es el uso de un modelo híbrido ARIMA-ANN para predicción de horizonte a corto plazo. Se ocuparon datos de México provenientes de Baja California, Zacatecas y Quintana Roo en estos buscan probar que la combinación de ARIMA-ANN es una buena opción; en donde el modelo ARIMA representa la parte lineal de las series y la ANN considera la parte no lineal. Se obtuvo buenos resultados predictivos de esta combinación.

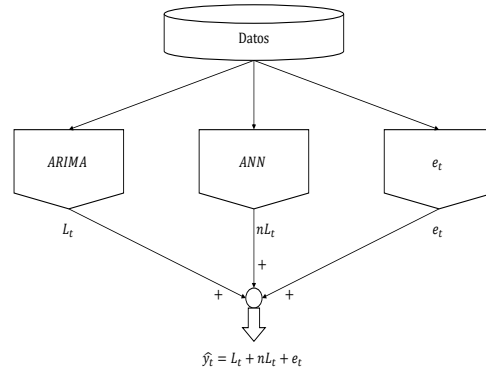


Figura 5.1: Modelo propuesto por Cadenas et al. 2010

El modelo realizado por Liu et al. 2012.[63]. Es una comparativa que muestra dos modelos híbridos; el primero es ARIMA con la unión de una red neuronal y el segundo combina ARIMA junto a un filtro de Kalman. Los datos son provenientes de una granja de energía eólica localizada China, en donde el horizonte de predicción no es mostrado, pero se obtienen buenos resultados de predicción en la combinación del modelo híbrido ARIMA-Kalman.

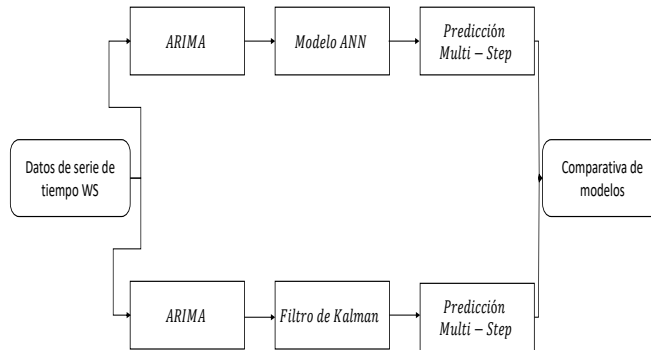


Figura 5.2: Modelo propuesto por Liu et al. 2012

El siguiente trabajo realizado por Yan-Fei et al. 2015 [64]. Es un modelo propuesto para una predicción a muy corto plazo que busca resolver un problema de viento en los trenes de japon, la primera parte es descomponer las series de tiempo de la velocidad del viento mediante la descomposición empírica (EMD por sus siglas en inglés Empirical Mode Decomposition), esto lo que hace es descomponer la serie en series de baja y alta frecuencia de una forma muy parecida a la EWT y después crear modelos recursivos ARIMA, una vez obtenidos estos se procede a obtener las predicciones de tales modelos para después hacer una inversión para regresar a una serie original y obtener la predicción.

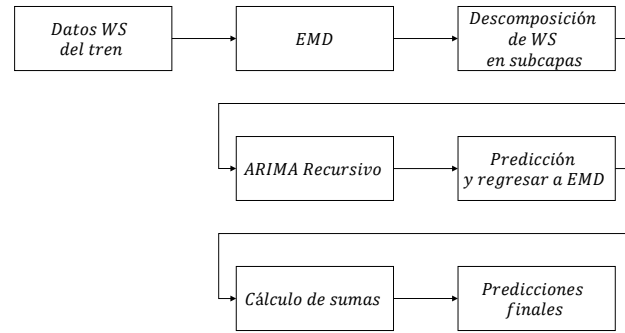


Figura 5.3: Modelo propuesto por Yan-Fei et al. 2015

El modelo realizado por Wang et al. 2015 [65] es una propuesta para una predicción a corto plazo, son dos parques eólicos localizados en china. Se realiza un pre-procesamiento de la señal mediante una transformada Wavelet (EWT por sus siglas en inglés Empirical Wavelet Transform), posteriormente, se hace una evaluación de distintos métodos estadísticos de predicción como: ARIMA, LSSVM, ELM y SVM. Cada salida de estos modelos se combinará en una última etapa, mediante un modelo de probabilidad gaussiana GPR para predecir la distribución de la velocidad del viento futura. El modelo propuesto muestra que supera a todos los modelos en solitario (ARIMA, LSSVM, ELM y SVM) sin utilizar una primera etapa de EWT.

Otro modelo referente realizado por Liu et al. 2015 [64]. Es una propuesta para una predicción a muy corto plazo que busca resolver un problema de viento en los trenes de japon. La primera parte es descomponer las series de tiempo de la velocidad del viento mediante la descomposición empírica (EMD por sus siglas en inglés Empirical Mode Decomposition). El propósito del EMD es descomponer la serie en sub-series de baja y alta frecuencia de una forma muy parecida a la EWT y posteriormente, crear modelos recursivos de ARIMA. Una vez obtenidos se procede a obtener las predicciones de los modelos para después hacer una inversión para regresar a una serie original y obtener la predicción.

El artículo realizado por Aasim et al. 2015 [66]. Utiliza los datos de la estación Malin Head en Irlanda con los modelos ARIMA y WT-ARIMA. En el objetivo es pronosticar la velocidad del viento y proponer un nuevo modelo ARIMA basado en WT (del inglés Wavelet) repetido (RWT-ARIMA), que mejora la precisión del modelo para el pronóstico de la velocidad del viento a muy corto plazo. Se realizó una comparación del modelo RWT-ARIMA propuesto con los modelos ARIMA y WT-ARIMA. Los modelos fueron sometidos a varias escalas de tiempo de pronóstico, como 1 min, 3 min, 5 min, 7 min y 10 minutos.

El siguiente modelo tiene bases similares a [64], realizado por Yunus et al. 2016 [41]. En el cual se presenta un modelo híbrido con predicción a corto plazo. En el que se aplica una descomposición de frecuencias, donde la serie de tiempo se separa en componentes de alta y baja frecuencia, una vez se obtiene esta separación, se procede a construir un modelo ARIMA de las series divididas. Al obtener la predicción de los modelos ARIMA realizados con las sub-series, estas deben invertirse a través de un tratamiento de frecuencias para obtener una sola serie. Los resultados de predicción de este proceso deben alcanzar un error aceptable y de no ser así se repite el mismo procedimiento hasta alcanzar un error mínimo.

Este trabajo es uno subsecuente de Cadenas et al. 2016 [7]. Esta nueva propuesta se basa en un modelo multivariante desarrollado por una red neuronal artificial exógena autorregresiva no lineal (NARX). Se comparó con un modelo ARIMA para datos de México ubicados en los estados de Hidalgo y Oaxaca. Es una predicción de horizonte a corto plazo en que el modelo NARX muestra una mejora respecto a un modelo ARIMA de un solo conjunto de datos.

El siguiente trabajo es una comparativa que muestra como se adaptan los modelos más usados en la literatura a los horizontes de tiempo. Nair et al. 2017 [67]. Realizó un artículo que muestra los diferentes modelos de predicción como lo son redes neuronales, ARIMA y la combinación de ambos ARIMA-ANN, para distintos horizontes de tiempo y comprobar en qué horizonte estos modelos son más aptos y con una mejor predicción. El resultado comprueba que los modelos híbridos tienen mejores resultados. Las series de tiempo fueron adquiridas de 3 sitios distintos en Tamil Nadu, India.

En este artículo Dimitru et al. 2019 [68]. Comparan dos modelos muy usados en el área de la predicción de energía eólica, como son el modelo ARIMA y la FFANN. Los datos provienen de una serie de tiempo de producción de electricidad, medidos en todo el mes de marzo 2018 del operador de sistema eléctrico nacional de Rumania. El objetivo es mostrar la capacidad de ambos modelos y realizar una comparativa para una predicción a corto plazo. Los resultados muestran que ARIMA obtiene menos error en las métricas de medición que una red neuronal. Sin embargo, en un segundo intento se realizó una segunda configuración de FFANN (del inglés FeedForward Artificial Neuronal Network), para realizar una mejora sobre el mismo modelo de ANN. El resultado demuestra que una simple reconfiguración de la red puede mejorar el modelo de la predicción por encima de un 15 % en las métricas de medición de errores.

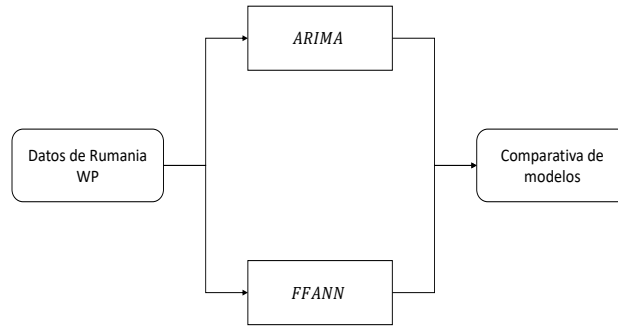


Figura 5.4: Modelo propuesto por Dimitru et al. 2019

Este artículo realizado por Niu et al. 2020 [54]. Adoptan un nuevo modelo de pronóstico híbrido que combina la descomposición modal empírica (EMD) y la red neuronal de regresión general (GRNN) optimizada por el algoritmo de optimización de la mosca de la fruta (FOA). En este nuevo modelo, las series originales de velocidad del viento se descomponen primero en una colección de funciones de modo intrínseco (FMI) y un residuo. El error porcentual absoluto medio de los resultados del pronóstico en dos casos es respectivamente 8,95 % y 9,87 %, lo que sugiere que el enfoque híbrido supera a los modelos comparados, lo que brinda orientación para el pronóstico futuro de la velocidad del viento.

Este artículo propuesto por Zhang et al. 2021 [69]. Desarrolla un modelo autorregresivo dinámico adaptativo (ARDA) basado en la mejora del modelo autorregresivo (AR). El método de estimación de parámetros fijos del modelo AR se mejora en el modelo propuesto a un método de estimación de parámetros paso a paso dinámicamente adaptable. Los coeficientes del modelo se actualizan de forma adaptativa en función de las características de los datos de energía eólica, lo que mejora la precisión del modelo propuesto. La precisión de predicción del modelo propuesto se mejora aún más mediante la función residual. El modelo se adapta bien a datos de energía eólica con diferentes grados de volatilidad. El modelo ARDA y otros dos modelos se probaron utilizando datos de energía eólica estacionaria y fluctuante (unidad: segundos), y se compararon los resultados de predicción de energía eólica en diferentes longitudes de paso de pronóstico. El modelo ARDA es más preciso, con una tasa de cálculo más rápida y una mejor adaptabilidad dinámica a las fluctuaciones de datos que los modelos ARIMA y LSTM.

5.2. Modelo híbrido múltiple ARIMA

Está comprobado que modelo ARIMA es efectivo para el pronóstico de series temporales de un solo conjunto de datos. No obstante, cuando faltan datos, hay ruido o los conjuntos de entrenamiento contienen pocos datos, los modelos ARIMA no son efectivos. El modelo múltiple ARIMA busca mejorar estas deficiencias utilizando más de un conjunto de datos, para crear un modelo ARIMA por cada conjunto y unir todas sus características mediante una red neuronal. Mejorando la precisión y robustez del modelo ARIMA original.

5.2.1. ARIMA múltiple

Cuando se tienen l conjuntos de diferentes datos

$$\{y_t^\sigma\}, \quad \sigma = \{1, 2, \dots, l\}$$

Es posible entrenar l modelos ARIMA $M_1 \cdots M_l$ usando el enfoque de modelado ARIMA para determinar (p, q, d) para cada conjunto de datos σ , y estimar los parámetros de cada (p, q, d) -orden el modelo ARIMA usados en (3.42). El vector de parámetros θ_i está asociado a un modelo ARIMA M_i . Los múltiples ARIMA se presentan como:

$$ARIMA_{\theta^i}(p^i, q^i, d^i), \quad i \in \sigma \quad (5.1)$$

El modelo $ARIMA_{\theta^i}(p^i, q^i, d^i)$ puede predecir los conjuntos de datos $\{y_t^i\}$.

5.2.2. Transferencia de aprendizaje en múltiples modelos ARIMA

Si hay incertidumbres en los conjuntos de datos de entrenamiento $\{y_t^\sigma\}$ o se tienen pocos datos, los modelos de predicción no pueden dar buenos resultados. Utilizando la propuesta de aprendizaje por transferencia y modelos múltiples se busca resolver estos problemas.

La propiedad fundamental del aprendizaje por transferencia es que puede retener y reutilizar el conocimiento aprendido previamente. Consiste en modelos pre-entrenados, donde los pesos finales de estos modelos se utilizan como condiciones iniciales del entrenamiento secuencial. Esto se puede adoptar como un método común para entrenar redes neuronales con menos información. Cuando los datos de entrenamiento son pocos y no se sienten completos, podemos tomar datos de otros conjuntos de datos similares. En este modelo nos inspirándonos en esta idea, proponemos el siguiente enfoque de aprendizaje por transferencia que se muestra en la siguiente ecuación, donde Λ_a y Λ_b son dos dominios relacionados y k se refiere a cada uno de los datos contenidos en ellos

$$\begin{aligned} \Lambda_a &= Y_a(k) = [y_a(k-1), y_a(k-2), \dots, y_a(k-n)] \\ \Lambda_b &= Y_b(k) = [y_b(k-1), y_b(k-2), \dots, y_b(k-n)] \end{aligned} \quad (5.2)$$

Suponemos que existe alguna relación (explícita o implícita) entre estos dominios, luego podemos usar el conocimiento de Λ_b para compensar la efectividad de los datos faltantes en Λ_b , es decir, podemos usar $\{\Lambda_a, \Lambda_b\}$ para entrenar el modelo M_a o podemos usar Λ_b para entrenar previamente el modelo M_a , luego usamos Λ_a para entrenar con precisión el modelo M_a , pero lo más importante para un aprendizaje exitoso es que el dominio con más información transfiere características al dominio con datos pequeños. En el caso general, tenemos l modelo de predicción y conjuntos de datos $\{\Lambda_i, \dots, \Lambda_l\}$, cada tarea de predicción se puede representar mediante un modelo ARIMA

$$\hat{y}_i(k) = ARIMA_i[(\hat{y}_i(k-1), \hat{y}_i(k-2), \dots, \hat{y}_i(k-n))] \quad (5.3)$$

dónde $i = \{1, \dots, l\}$ $i \in \sigma$. Sea S un conjunto cerrado y acotado que representa un espacio de parámetros de dimensión finita

$$ARIMA_i(D_i)$$

donde D_i es el i -ésimo conjunto de datos, $ARIMA_i$ es i -ésimo modelo ARIMA.

El vector de parámetros del modelo $\theta^i \in S$, ya que todos los modelos se pueden parametrizar de la misma forma. Definimos θ^{i*} como los parámetros subóptimos del i -ésimo modelo ARIMA. Existe una región de límite cerrado Ω , tal que $\theta^{i*} \in \Omega$, entonces

$$\theta^{i*} \in S$$

Esto significa que podemos combinar todos los M_i , $i = 1, \dots, l$ de manera adecuada, aunque sus dominios sean diferentes.

5.2.3. ARIMA múltiple combinado con redes neuronales

Las redes neuronales son modelos basados en datos. Son más flexibles que ARIMA, pero la precisión del modelado para una serie de tiempo es peor que ARIMA. La combinación de redes neuronales y ARIMA puede utilizar ambas ventajas de estos dos modelos. Para el modelo ARIMA único, existen varios tipos de compensación, como la compensación directa [70].

$$y = ARIMA(D) + y_{NN}$$

Aquí, la red neuronal se usa para aprender el error de modelado de ARIMA(D). O combinación de productos [71].

$$y = ARIMA(D) * y_c, \quad y_c = \frac{y_{NN}}{ARIMA(p, q, d)}$$

donde $ARIMA(p, q, d)$ es el resultado previo al entrenamiento. O conexión serie [36]

$$y_{NN} = \sum_{j=1}^{n_2} v_j \phi\left(\sum_{o=1}^{n_1} w_{ij} A_{t-o}\right), \quad A_{t-o} = ARIMA(p, q, d) \quad (5.4)$$

Para la propuesta de este modelo se utiliza una red neuronal que combina varios modelos ARIMA. Se construyeron 2 estrategias para combinarlos: 1) ARIMA múltiple con fusión de redes neuronales; 2) ARIMA múltiple con compensación de red neuronal.

Combinación de modelos ARIMA usando redes neuronales

El modelo de red neuronal más popular es el perceptrón multicapa (MLP). Un MLP de dos capas se puede escribir como

$$\hat{y}_i(k+1) = \sum_{j=1}^n W_j \phi \left[\sum_{o=1}^l V_{oj} \hat{y}_i(k-o+1) \right] \quad (5.5)$$

donde $\hat{y}_i(k+1)$ es la salida del MLP, ϕ es la función activa no lineal, V_{oj} y W_j son los pesos, n_1 y n_2 son el número de neuronas en las dos capas.

Para usar una red neuronal para combinar diferentes modelos ARIMA, se requiere el aprendizaje de transferencia discutido anteriormente, es decir, diferentes conjuntos de datos con sus propios modelos ARIMA pueden ayudar a otro modelo ARIMA a mejorar su precisión de predicción. El esquema de combinación de diferentes modelos ARIMA utilizando una red neuronal se muestra en la Figura 5.5.

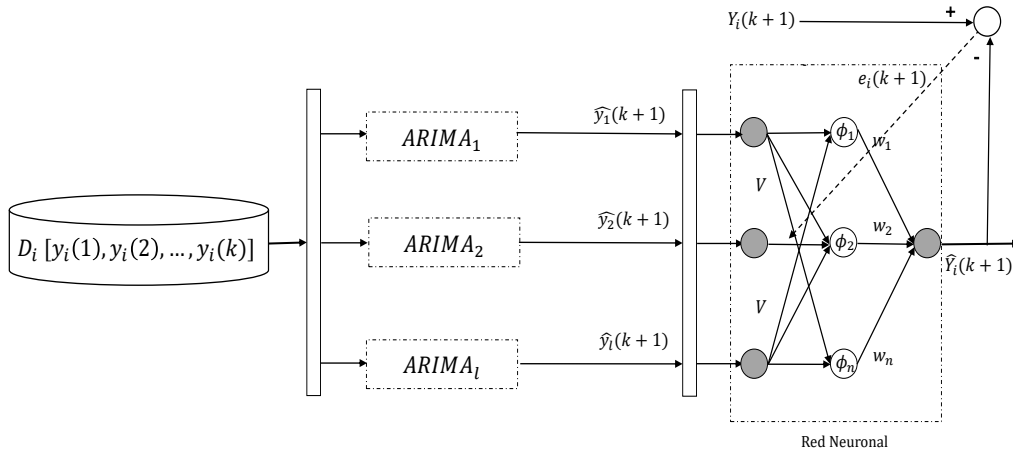


Figura 5.5: Esquema de combinación del modelo ARIMA y ANN.

La salida de los múltiples modelos ARIMA es

$$\hat{y}_{ni}(k+1) = \sum_{j=1}^n W_j \phi \left[\sum_{o=1}^l V_{oj} A_o(k) \right] \quad (5.6)$$

donde $A_o(k)$ es el resultado de cada modelo ARIMA obtenido en la figura 3.1,

$$A_o(k) = ARIMA_o(D_i), \quad o = 1 \dots l,$$

La tarea principal es entrenar (5.6) con el error de entrenamiento

$$e_i(k+1) = y_i(k+1) - \hat{y}_{ni}(k+1)$$

tal que

$$[V^*, W^*] = \arg \min_{V, W} \sum e_i^2(k+1) \quad (5.7)$$

dónde $V = \{V_{oj}\}$, $W = \{W_j\}$. Con el algoritmo de retropropagación se obtiene $[V^*, W^*]$. Donde $ARIMA_j(D_j)$ es entrenado por su propio conjunto de datos D_j como en la figura 5.5, posteriormente la red neuronal NN_i es entrenada por D_i , tal que $\hat{y}_{ni}(k+1)$ de (5.6) es mejor que $\hat{y}_i(k+1)$

$$\hat{y}_i(k+1) = ARIMA_i(D_i)$$

La diferencia entre el método ARIMA múltiple propuesto y los métodos híbridos que presenta la literatura [[72],[71]], es que las entradas a NN de (5.6) son la salida de modelos ARIMA "múltiples", mientras que las entradas a NN en [[72],[71]] son de uno solo modelo ARIMA. Una ventaja del modelo propuesto es que posible extraer más características aplicando las técnicas de aprendizaje por transferencia en múltiples conjuntos de datos.

Compensación de modelos ARIMA usando red neuronal

Este modelo se modificó del propuesto original a través de los experimentos realizados, debido a que no todos los modelos se adaptaron de forma correcta. Por tanto, el esquema considera usar una compensación a la red neuronal, con el motivo de que aprenda del error de predicción del modelo ARIMA, vea la Figura 5.6.

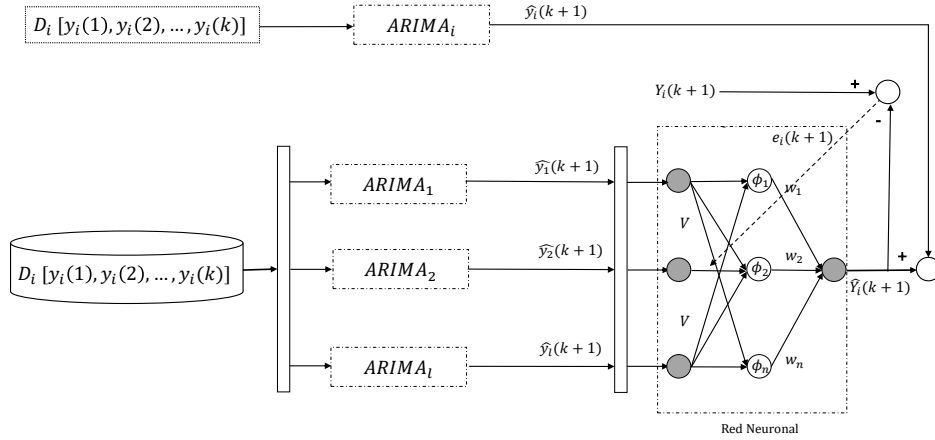


Figura 5.6: Esquema de combinación del modelo ARIMA y ANN con compensación.

Después del primer paso de entrenamiento como en la Figura 3.1, la predicción el error del modelo ARIMA es

$$e_i(k+1) = y_i(k+1) - ARIMA_i(D_i)$$

La salida de los múltiples modelos ARIMA es la misma que (5.6), pero $A_o(k)$ es

$$A_o(k) = ARIMA_o(D_o), \quad o = 1 \cdots l, \quad o \neq i$$

El error de entrenamiento es

$$\hat{e}_i(k+1) = e_i(k+1) - \hat{y}_{ni}(k+1)$$

La minimización del error de entrenamiento es

$$[V^*, W^*] = \arg \min_{V, W} \sum \hat{e}_i^2(k+1) \quad (5.8)$$

Después de entrenar la red neuronal, la predicción del i-ésimo modelo es

$$\begin{array}{l} ARIMA_i(D_i) + \hat{y}_{ni}(k+1) \\ o \quad ARIMA_i(D_i) + \hat{e}_i(k+1) \end{array}$$

Capítulo 6

Pronóstico de energía eólica con modelo híbrido múltiple ARIMA

En este capítulo se describen los conjuntos de datos con más detalle; la transformación de cada conjunto a serie de tiempo utilizando como unidad de tiempo los días respecto a la energía eólica generada en Giga Watts, la comprobación de la prueba ADF para verificar si una serie es estacionaria o en su defecto en que diferencia se vuelve estacionaria, la obtención de los parámetros (p, d, q) para construir los modelos ARIMA, la unión de estos modelos con la red neuronal, la elección de la red neuronal y los códigos en Python y MATLAB que beneficiaron a cumplir las tareas anteriores.

6.1. Conjuntos de datos

La parte de los datos fue y es una problemática que surge del país; los datos respecto a energía eléctrica no se pueden conseguir de ninguna forma que no sea pasando por la Comisión Federal de Electricidad (CFE) de México. Esto complicó uno de los objetivos principales que era recabar datos de granjas eólicas en México, algunos artículos [20], [7] han podido trabajar con estos datos haciendo predicción de la velocidad del viento, pero mencionan que los datos son propiedad de CFE, por tanto, no provienen de una base que pueda ser utilizada. Por ello decidimos plantear el modelo con conjuntos de acceso libre y en un futuro incluir datos de México.

Se obtuvieron 3 bases de datos de distintos sitios: Kaggle (plataforma web de datos y competencias online), Alemania y California. Posteriormente se realizó el procesamiento de los conjuntos para convertirlos en un formato en días de energía eólica generada. A continuación, se describirá cada una de las bases de datos antes de realizar la limpieza de anomalías y la conversión a serie de tiempo de días respecto a energía generada.

Datos Kaggle.

Conjunto de datos obtenido en la competencia *Global Energy forecasting Competition 2012*. Los valores de las series de tiempo son recolectados de 7 granjas eólicas en un periodo comprendido entre julio 2009 y junio del 2012, para las pruebas se tomaron los datos de la primera granja en un periodo comprendido entre 1 de julio 2009 hasta el 31 de diciembre de 2010. El formato de la fecha del conjunto es de la forma 2009071102 siendo los primeros 4 números el año (2009), los dos siguientes el mes (07), los consiguientes dos el día (11) y los últimos la hora (02) presentada en un formato de 24 horas. En la figura 6.1 se observa una visualización de los datos y una flecha que nos señala la variable de energía generada que se utilizó, cada columna representa una granja distinta y las mediciones de energía eólica que obtuvo en los distintos tiempos. Mediante una sumatoria de las horas en cada fecha se obtuvo la serie de tiempo de los datos de Kaggle. Información a detalle de los datos en [73].

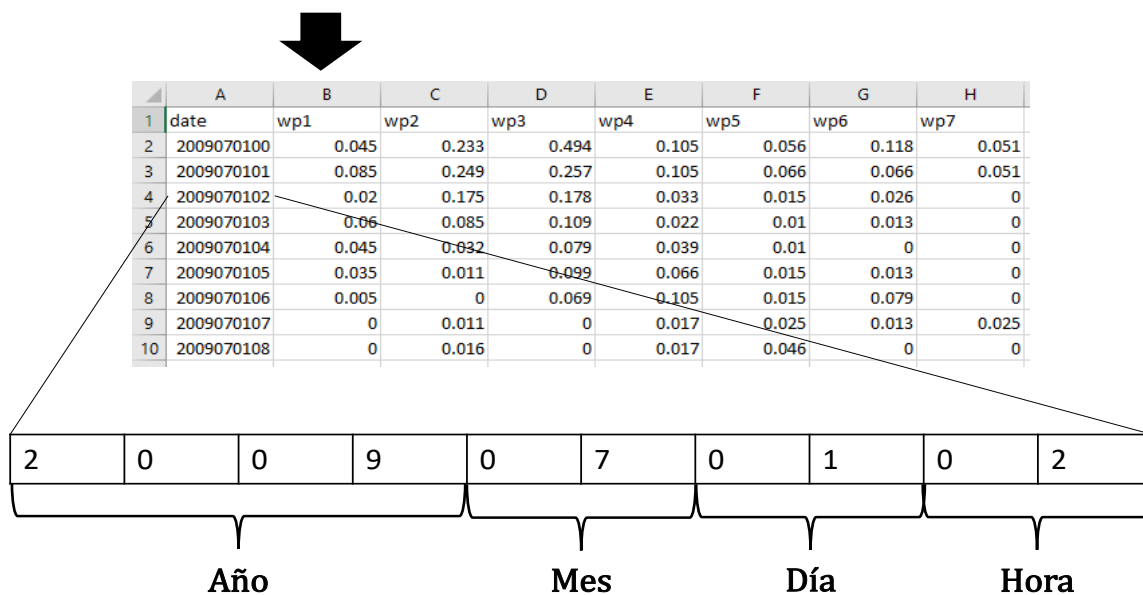



Figura 6.1: Datos de Kaggle y visualización del formato de fechas.

Datos California.

Datos obtenidos en la página de la ISO (Independent System Operator) California, compañía que administra el flujo de electricidad en un 80 % de líneas de transmisión de alta tensión y larga distancia de California. Los datos se presentan en columnas de fecha, hora (formato de 24 horas), junto con los siguientes tipos de energía: biogás, geotérmica, biomasa, hidro, eólica total, solar fotovoltaica y solar térmica. En la figura 6.2 se observa el formato de los datos de tiempo con los que se creó una sumatoria de las horas para conformar los días para la serie de tiempo, mientras que una flecha nos indica la columna elegida de la energía eólica. El registro del periodo tomado para la serie de tiempo fue del 1 de septiembre de 2011 hasta 31 de agosto del 2012. Información a detalle de los datos en [74].



	A	B	C	D	E	F	G	H	I	J
1	TIMESTAMP	BIOGAS	BIOMASS	GEOTHERMA	Hour	SMALL HYDR	SOLAR	SOLAR PV	SOLAR THERM	WIND TOTAL
2	7/21/2011 0:00	177	372	984	1	516	0			1607
3	7/21/2011 1:00	176	373	984	2	509	0			1619
4	7/21/2011 2:00	177	371	985	3	506	0			1660
5	7/21/2011 3:00	177	374	985	4	504	0			1633
6	7/21/2011 4:00	177	377	984	5	508	0			1520
7	7/21/2011 5:00	177	371	980	6	510	0			1406
8	7/21/2011 6:00	176	372	979	7	521	10			1391
9	7/21/2011 7:00	176	378	977	8	524	84			1341
10	7/21/2011 8:00	176	380	975	9	526	169			1220
11	7/21/2011 9:00	175	381	973	10	528	254			1025
12	7/21/2011 10:00	175	383	971	11	530	339			798

7 / 21 / 2011	4:00
---------------	------

Fecha
Hora

Figura 6.2: Datos de California y visualización del formato de fechas.

Datos Alemania.

Conjunto de datos obtenido del sitio netztransparenz, contiene información publicada conjuntamente por 4 operadores de sistemas de transmisión alemanes. Las granjas eólicas pertenecen a las siguientes empresas: Tennet, 50Hertz, TransnetBW y Amprion. El conjunto seleccionado para los experimentos fue Tennet. Los datos de generación de energía son tomados en intervalos de quince minutos dando un total de 96 mediciones por día. De esta forma al hacer la sumatoria de las 96 mediciones se obtuvieron las mediciones para cada fecha correspondiente. En la figura 6.3 se observa que la primera columna corresponde a las fechas y cada fila a partir de la segunda columna son los pasos de medición del tiempo que fueron sumados para así obtener una serie de tiempo en días. Información a detalle de los datos en [75].

	A	B	C	D	E	F	...	CP	CQ	CR	CS
1	Date	0:00:00	0:15:00	0:30:00	0:45:00	1:00:00	...	23:00:00	23:15:00	23:30:00	23:45:00
2	23/08/2019	9.68	10.16	10.94	11.39	12.09		71.26	71.95	69.97	69.58
3	24/08/2019	67.94	67.52	64.48	64.78	65.75		99.04	96.57	94.75	91.73
4	25/08/2019	89.38	88.95	85.13	82.74	81.46		37.6	35.46	34	33.47
5	26/08/2019	33.71	31.84	32.75	33.33	32.02		57.53	58.15	55.76	54.34
6	27/08/2019	52.65	47.95	44.61	43.8	41		55.51	55.96	56.48	55.19
7	28/08/2019	53.12	49.72	47.85	47.76	47.25		22.78	20.54	17	12.47
8	29/08/2019	11.28	10.49	11.65	12.96	14.57		8.5	8.87	8.47	8.44
9	30/08/2019	8.47	8.45	9.19	9.7	9.5		14.95	15.51	15.66	15.99
10	31/08/2019	16.14	16.98	17.51	17.81	16.91	...	164.38	156.88	142.93	121.01

31 / 08 / 2019	0:15:00
----------------	---------

Fecha

Tiempo

Figura 6.3: Datos de Tennet-Alemania y visualización del formato de fechas.

6.2. Pre-procesamiento de datos

Respecto al horizonte de predicción se decidió tomar una predicción de largo plazo, debido a la naturaleza de los conjuntos. Varios artículos mencionan que los modelos estadísticos y basados en aprendizaje automático funcionan mejor con predicciones a corto plazo, esto quiere decir que las predicciones tienen que ir en la medición de minutos (revisar 2.2), en nuestro caso optamos por una medición en días debido a que las bases originales tenían perdidas de mediciones en bastantes horas del día, por tanto, para hacer más homogéneos los datos realizamos las sumatorias de todas la mediciones y sintetizamos las series respecto a días.

Para el procesamiento de los datos utilizamos el lenguaje Python versión 3, en el siguiente código 6.2 se explica la normalización para los conjuntos de datos y así crear cada una de las series estacionarias.

Una vez obtenidos los tres conjuntos de datos seleccionados (Kaggle (Granja 1), California, Alemania (Tennet)) fueron normalizados alrededor de los valores $[0, 20]$ para evitar anomalías y redundancias entre estos, se utilizó la ecuación de normalización siguiente:

$$X_{norm}(i) = \frac{x_i - \min(x)}{\max(x) - \min(x)} \quad (6.1)$$

Donde x_i es cada dato de la serie de tiempo, mientras que $\min(x)$ está denotado por 0 y $\max(x)$ está denotado por 20, para así obtener la relación a la que pertenece cada dato entre las cotas propuestas.

```

1  from sklearn.datasets import load_iris
2  from sklearn.preprocessing import MinMaxScaler
3  from sklearn import preprocessing
4  import numpy as np
5  import matplotlib.pyplot as plt
6  import pandas as pd
7
8  serie      = pd.read_csv('datos.csv', parse_dates=True, index_col=0)
9  minmaxsca  = preprocessing.MinMaxScaler(feature_range=(0,20))
10 x_scaler   = minmaxsca.fit_transform(serie)
11 df        = pd.DataFrame(x_scaler)
12 serieenerg = df[0]
```

Como se puede observar en código anterior, en la línea 8 tomamos los datos de cada serie en un formato .csv, dónde posteriormente en el código pasamos a realizar la normalización con el procedimiento de Python x_scaler y una vez lo movemos a un objeto podemos visualizar su contenido y graficar cada una de las series.

Las siguientes gráficas muestran los conjuntos de datos normalizados en el margen antes mencionado, todos los datos se componen por fecha en días en el eje de las abscisas, mientras que en el eje de las ordenadas muestra la energía generada en Giga Watts (GW).

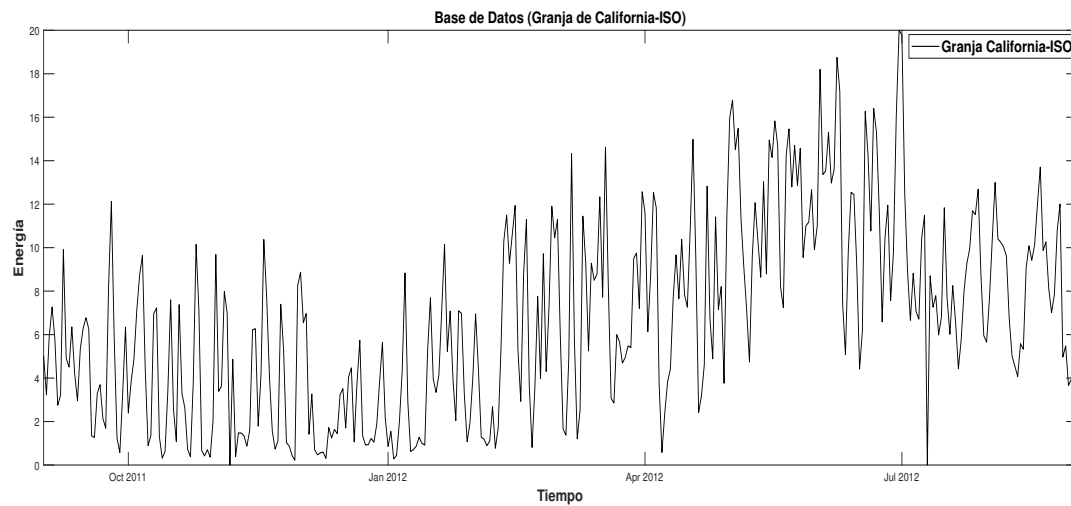


Figura 6.4: Serie de tiempo normalizada de los datos California del periodo comprendido entre 1 de septiembre de 2011 hasta 31 de agosto del 2012.

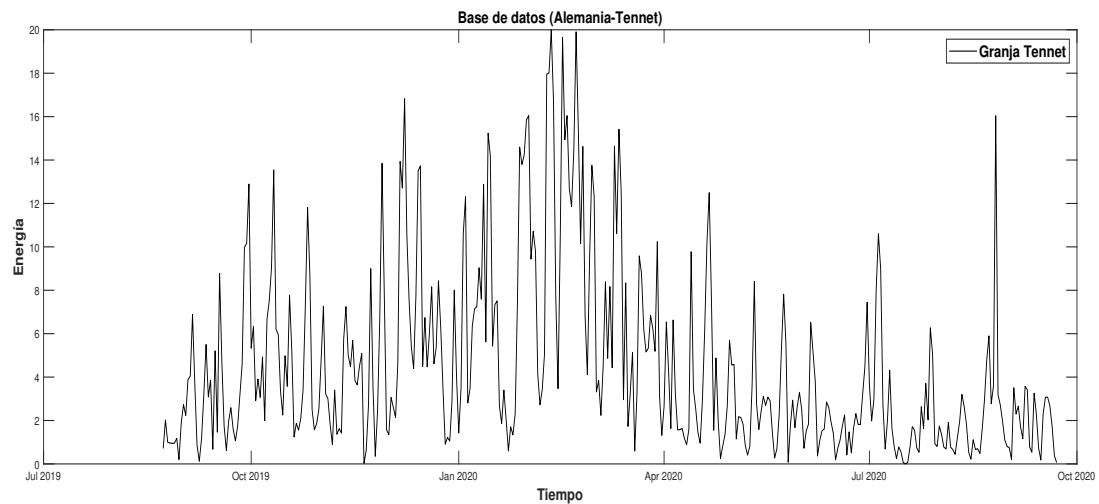


Figura 6.5: Serie de tiempo normalizada de los datos Alemania-Tennet del periodo comprendido entre 23 de agosto de 2019 hasta 22 de septiembre del 2020.

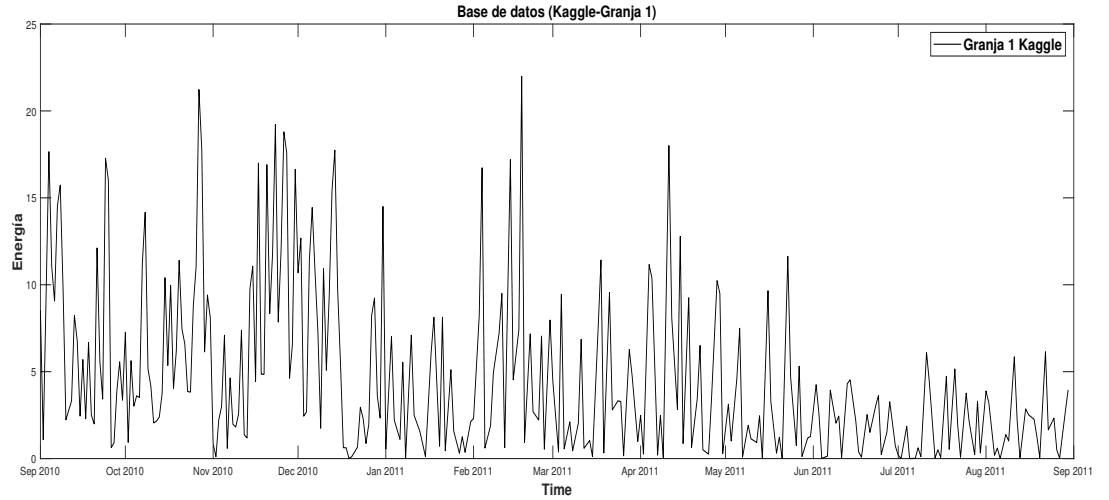


Figura 6.6: Serie de tiempo normalizada de los datos Kaggle (Granja 1) del periodo comprendido entre 1 de septiembre de 2010 hasta 30 de agosto del 2011.

Al poder visualizar como se han creado nuestras series de tiempo en la forma correcta, se debe analizar si las series son estacionarias o en qué diferencia la serie puede convertirse en estacionaria.

6.3. Prueba estacionaria

Es esencial para el modelo ARIMA conocer si la serie de tiempo es o no estacionaria, dado que uno de los parámetros del modelo ARIMA requiere de saber de cuantas diferencias requiere una serie para convertirse en estacionaria o si la serie ya lo era, el parámetro d toma el valor de 0.

A continuación, se presenta el análisis que se obtuvo por medio del método Dickey-Fuller aumentado (ADF en inglés Augmented Dickey-Fuller).

Serie	ADF Estadístico	p-valor	1 %	5 %	10 %	H0	d
Kaggle	-2.153723	0.22346750	-3.449	-2.870	-2.571	not stationary	0
Kaggle	-7.903060	4.1355332e-12	-3.449	-2.870	-2.571	stationary	1
Germany	-2.898682	0.0455013	-3.447	-2.869	-2.569	stationary	0
California	-1.226807	0.661958	-3.449	-2.870	-2.571	not stationary	0
California	-9.737932	8.649381e-17	-3.449	-2.870	-2.571	stationary	1

Tabla 6.1: Resultado de la prueba Dickey-Fuller en los conjuntos de datos

El código 6.3 da como resultado la respuesta de la prueba ADF para las diferentes series de tiempo:

```

1 resultado = adfuller(serieenerg)
2 print('ADF Statistic: %f ' % result[0])
3 print('p-value: %f ' % result[1])
4 print('Critical Values: ')
5 for key,value in result [4].items():
6     print('\t%s: %.3f' % (key, value))
7 if resultado[0] < resultado [4][ "5%"]:
8     print("Es estacionaria")
9 else:
10    print("No estacionaria")

```

Para explicar el código anterior se utiliza la función `adfuller` directamente sobre nuestra serie de tiempo normalizada, esta nos arrojará el resultado de la prueba en la cual buscamos comparar con nuestros valores críticos en los distintos porcentajes de aceptación. Si el valor del ADF es menor que el valor crítico al 5 % la serie puede determinarse estacionaria, se puede considerar también el valor del 10 % y el 1 % pero estos sirven como un límite superior e inferior que delimitan el resultado de la prueba.

Con base en la subsección 2.4.1 y con apoyo en la figura 2.5 se resume lo siguiente: Para determinar que la series son estacionarias, en la tabla 6.1, se pueden observar los valores críticos del 1 %, 5 % y 10 % deben compararse con el resultado de la prueba para determinar si la hipótesis de Dickey-Fuller se rechaza o se acepta, en caso de aceptar la hipótesis indicaría tener una raíz unitaria y por tanto la serie se determina como no estacionaria, mientras que, si se rechaza, la serie se determinará como estacionaria. Por tanto el conjunto de Kaggle es estacionario hasta la primer diferenciación debido a que en la primera prueba (diferenciación 0) obtiene un valor de -2.153, viendo todos los valores críticos y comparando, se puede determinar qué valor ADF (Kaggle) de la prueba es menor a todos estos, por tanto, la serie de tiempo necesita diferenciarse, así como también el conjunto de California presenta una tendencia similar, en la primera prueba el valor de ADF (California) es de -1.22, comparándolo con los valores críticos se obtiene que es menor a todos ellos, pero al diferenciar una primera vez logra superarlos, mientras que el conjunto de Alemania al obtener un valor del ADF en -2.898 al compararlo con el 1 % de valor crítico la prueba no sería válida, a pesar de ello, logra ser mayor que los otros valores críticos (5 % y 10 %) y por tanto, la serie queda determinada como estacionaria.

6.4. Entrenamiento de los modelos ARIMA

Los ARIMA de los 3 modelos en un inicio se trabajaron en la parte de Python y posteriormente fueron pasados a lenguaje R para comprobar un mecanismo de ensamble que surgió como prueba, para comprobar que se tuvieran bien los modelos ARIMA óptimos verificamos con la app Econometric Modeler de MATLAB que tuviéramos los mismos resultados de estimación de AIC y resultó tan conveniente que a partir del cálculo de los modelos ARIMA, todo lo demás fue desarrollado en MATLAB por la fácil unión que nos representó crear la red neuronal.

Para comenzar el entrenamiento de los modelos ARIMA se requiere especificar los tres parámetros de orden $(\mathbf{p}, \mathbf{d}, \mathbf{q})$, en donde el significado de cada uno de ellos se refiere a lo siguiente.

- \mathbf{p} es el orden del término AR (del inglés Autoregressive) que usa los valores pasados en la ecuación de regresión para la serie temporal.
- \mathbf{d} es el número de diferenciaciones requeridas para hacer que la serie de tiempo sea estacionario.
- \mathbf{q} es el orden del término MA (del inglés Moving Average) que representa el error del modelo como una combinación de términos de errores anteriores.

Para crear los modelos ARIMA de cada conjunto de datos se determinan los parámetros de \mathbf{p} , \mathbf{d} y \mathbf{q} . En el caso del parámetro \mathbf{d} fue calculado mediante el método de Dickey-Fuller Aumentado en el apartado anterior, los dos parámetros restantes requieren de un análisis de las gráficas de autocorrelación total (ACF Autocorrelation Function) y autocorrelación parcial (PAF Partial Autocorrelation Function).

Las siguientes gráficas determinan las pruebas de autocorrelación parcial y autocorrelación, las cuales determinan el número de p coeficientes de autoregresiones que necesita la serie de tiempo y el número de q coeficientes de media móvil para construir el modelado de las series. Las gráficas no devuelven un modelo en particular para cada serie de tiempo, su función es delimitar los parámetros del modelo ARIMA que mejor aproximan la serie de tiempo que se quiere modelar. Este método requiere generar un algoritmo de caminata aleatoria o una búsqueda a fuerza bruta para encontrar los mejores parámetros, para evitar el uso de un algoritmo de ese estilo, procedimos a ocupar el método de criterio de Aikake mencionado en la sección 3.3.1.

A continuación se describen las características que contienen cada una de las gráficas respecto a los parámetros p y q para cada serie de tiempo:

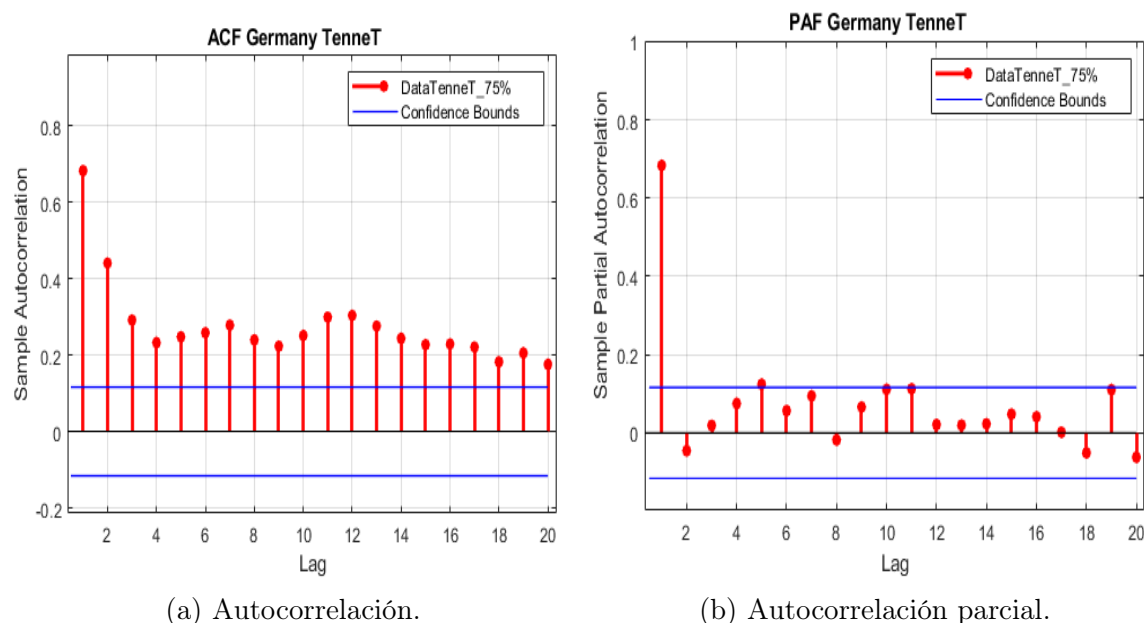


Figura 6.7: ACF/PCF Análisis para el conjunto de datos de Alemania.

La figura 6.7a muestra el análisis del ACF y determina los posibles candidatos para el parámetro q . El análisis del gráfico dictamina que las opciones para el parámetro son 1, 2, 3, 4 y 5. Para examinar el caso de p se selecciona la gráfica 6.7b, en donde se determina que los posibles candidatos son 1 y 2. Dando como resultado varias combinaciones para el conjunto de datos de Alemania

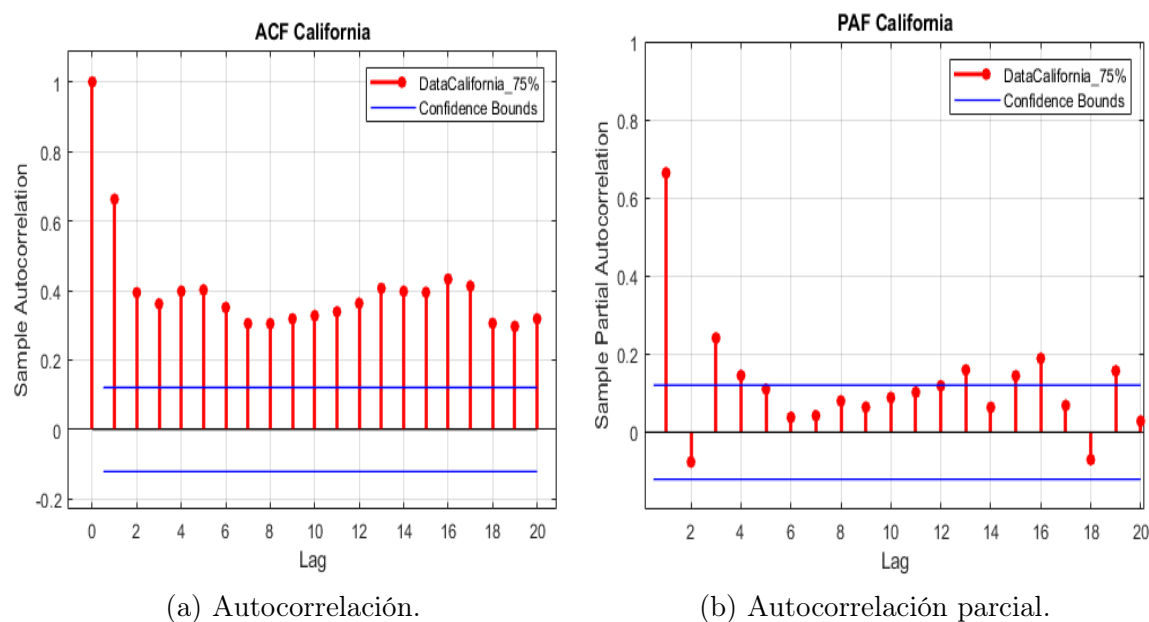


Figura 6.8: ACF/PCF Análisis para el conjunto de datos de California.

El procedimiento anterior tiene que repetirse para el conjunto de California. Con base en la figura 6.8a, al observar el gráfico ACF se determina que las opciones para q son 1, 2, 3, 4, 5 y 6, mientras que la gráfica 6.8b se utiliza para la elección de p y de esta forma los posibles candidatos son 1, 2, 3 y 4.

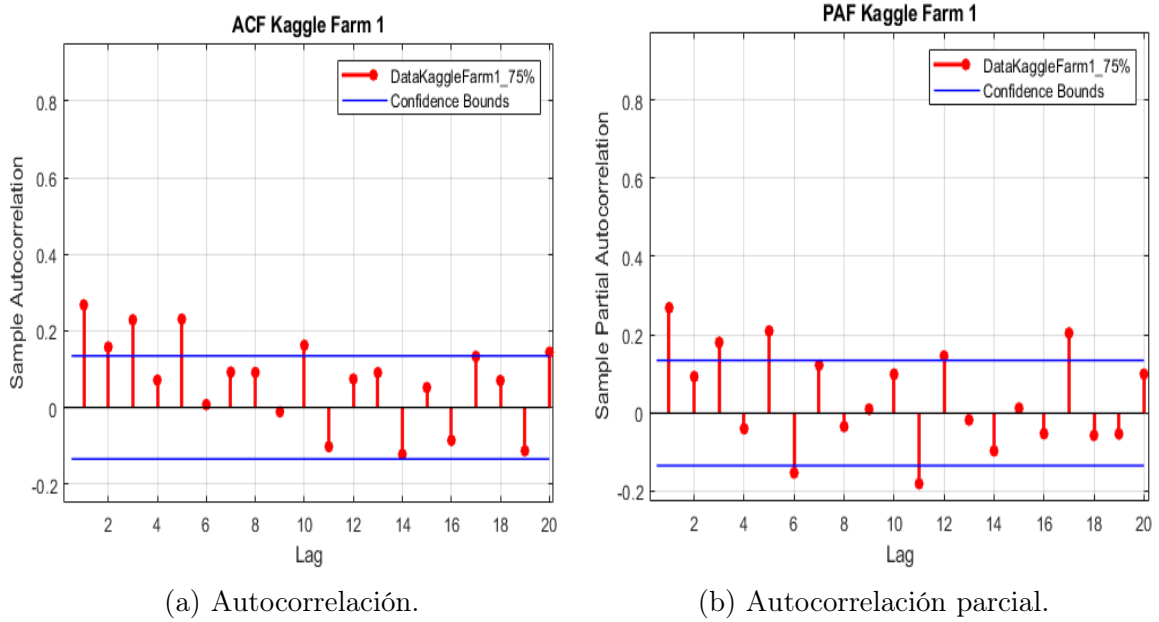


Figura 6.9: ACF/PCF Análisis para el conjunto de datos de Kaggle.

Reutilizando el procedimiento de los anteriores conjuntos, en la figura 6.9 correspondiente a Kaggle se procede a determinar los candidatos posibles para cada parámetro, al revisar el gráfico ACF 6.9a se determina que las opciones adecuadas son 1, 2 y 3, mientras la gráfica 6.9b de PAF determina que las mejores opciones para p son 1, 2 y 3.

La tabla 6.2 es un breve resumen de los parámetros candidatos que pueden construir los modelos ARIMA para las series de tiempo.

Serie	p	d	q
Alemania	1,2	0	1,2,3,4,5
California	1,2,3,4,5,6	1	1,2,3,4
Kaggle	1,2	1	1,2,3

Tabla 6.2: Resumen de los parámetros ARIMA candidatos obtenidos con las pruebas de ACF, PAF y la prueba Dickey-Fuller aumentada.

La tabla 6.2 acota el rango de búsqueda de los parámetros (p, d, q) para la selección de un modelo óptimo en cada conjunto de datos. Sin embargo, como ya se ha mencionado requiere de una búsqueda por fuerza bruta para medir la predicción de cada modelo, se descartó esa idea para utilizar el criterio AIC (Aikake Information Criterion).

Una vez evaluados los parámetros (p, d, q) y obteniendo todos los modelos posibles, procedemos a usar el AIC para compararlos. Las puntuaciones más bajas son mejores, y AIC penaliza a los modelos que utilizan más parámetros. Entonces, si dos modelos tienen la misma cantidad de variación, el que tenga menos parámetros tendrá una ponderación AIC más baja y será el modelo de mejor ajuste.

El código para encontrar el mejor modelo de parámetros es el siguiente:

```
1 modeloopt = auto_arima(serieenerg, trace=True, seasonal=True, stepwise=
    True, approximation=True, suppress_warnings=True)
```

Con la función `auto_arima` podemos determinar el modelo mejor adaptado a cada serie de tiempo, basándonos en el AIC obtenido.

Alemania[TenneTTSO]		California		Kaggle[Granja 1]	
Model	AIC	Model	AIC	Model	AIC
ARIMA(1,0,4)	1544.3	ARIMA(0,1,2)	1367.3	ARIMA(1,1,2)	1336.2
ARIMA(1,1,4)	1544.8	ARIMA(0,1,3)	1368.5	ARIMA(2,1,1)	1337.1
ARIMA(2,0,5)	1545.0	ARIMA(1,1,2)	1368.8	ARIMA(2,1,3)	1337.8
ARIMA(1,1,2)	1546.7	ARIMA(0,1,4)	1369.2	ARIMA(1,1,3)	1338.2
ARIMA(2,1,3)	1546.9	ARIMA(1,1,3)	1369.7	ARIMA(2,1,2)	1338.3
ARIMA(1,1,3)	1547.0	ARIMA(1,1,4)	1371.2	ARIMA(1,1,1)	1340.1
ARIMA(2,0,2)	1554.3	ARIMA(2,1,3)	1371.3		
ARIMA(1,0,2)	1554.8	ARIMA(6,1,2)	1376.4		

Tabla 6.3: AIC de los modelos ARIMA para los conjuntos de datos

La tabla 6.3 muestra AIC de los mejores modelos obtenidos de las combinaciones de parámetros de la tabla 6.2, organizados de menor a mayor, es por ello que se determina que el modelo ARIMA (1,0,4) es el mejor para el conjunto de Alemania, el modelo ARIMA (0,1,2) es la mejor opción para California y por último el modelo ARIMA (1,1,2) fue el seleccionado para Kaggle.

6.5. Predicción del modelo ARIMA múltiple

Para la elección de la red neuronal de primera instancia se utilizó la app de Regression Learner de MATLAB, que da la opción de optimizar una red neuronal con los datos de entrenamiento y encontrar un modelo optimizado basado en el error MSE calculado. A pesar de ello las combinaciones de redes neuronales que sugerían fueron combinaciones 3 capas con neuronas entre (1-100) en la primera capa y (1-10) neuronas en la segunda capa. Se probaron las combinaciones posibles sugeridas del software, pero ninguna de estas nos dio un resultado satisfactorio. Por ello se creó una red neuronal básica manteniendo las 3 capas sugeridas por el software, pero con un tamaño de neuronas alrededor de (1-10), dando una mejor respuesta la combinación de red de 3 neuronas en la capa de entrada, 2 en la capa oculta y 1 en la capa de salida.

A continuación, se presentan los resultados de predicción para el modelo propuesto, cabe destacar que la configuración de la red neuronal fue determinada por una configuración de 3 capas, en donde la capa de entrada son 3 neuronas, la capa oculta consta de 2 neuronas y la última capa que pertenece a la capa de salida consta de una sola neurona. Cada conjunto de datos fue dividido en dos partes: los datos de entrenamiento, que constan del 75 % en cada serie de tiempo y los datos de prueba que son el 25 % restante de los datos de cada conjunto.

6.5.1. Predicción del conjunto de prueba para granja California-ISO

Esta sección muestra los resultados obtenidos por el modelo ARIMA-múltiple, así como los resultados obtenidos por el ARIMA(0,1,2) para el conjunto de datos de California.

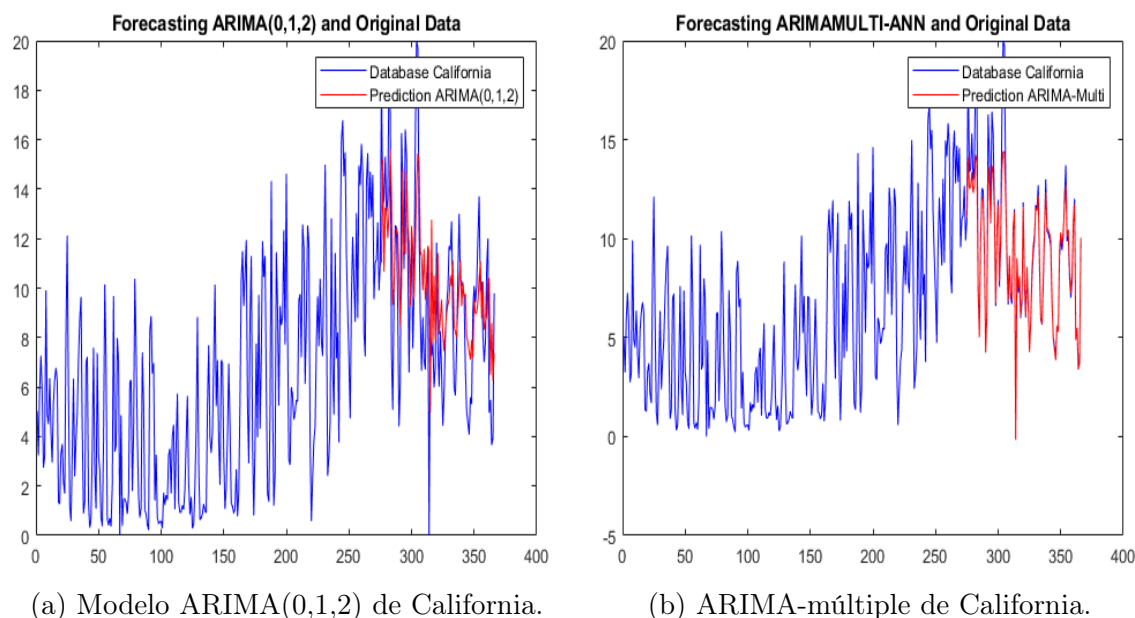


Figura 6.10: Resultado de predicción de los modelos con los datos de California.

La figura 6.10a es el resultado de los datos de prueba para ARIMA(0,1,2)-California óptimo, mientras que la figura 6.10 es la respuesta del modelo ARIMA-múltiple para California. Aunque se puede apreciar comparando las dos gráficas que la respuesta es mejor en 6.10b, se procede a analizar las métricas obtenidas entre estos dos modelos.

Modelo	MAE	SMAPE	RMSE	RSQ
ARIMA(0,1,2) California	2.4117	0.2676	3.139	0.3428
ARIMA-Multi California	0.6181	0.0715	1.263	0.8934

Tabla 6.4: Métricas para los modelos de California

La tabla 6.4 muestra una comparativa de las métricas para los datos de prueba aplicados en California. Cada una de las métricas en el modelo ARIMA-Múltiple presenta un error menor de casi el 50 % respecto al otro modelo. En el caso del MAE presenta una mejora del 74.37 % y la métrica nos indica que cuanto más cercano a 0, se obtiene un mejor rendimiento. Para la métrica SMAPE la mejora es del 73.28 %, esta métrica nos indica que cuando se acerca a 0 el rendimiento del modelo es de

buena predicción. La métrica RMSE se comporta de forma similar a la MAE, la mejora que presenta el modelo múltiple es de 59.76 % respecto al modelo ARIMA. La última métrica es el RSQ con una mejora del 55.06 %, en este caso cuando la métrica es cercana 1, el modelo se acopla mejor a los datos originales. Se puede observar que el modelo ARIMA-múltiple de California respecto a su ARIMA óptimo, no solo la respuesta gráfica, las métricas de errores también muestran que el ARIMA-Múltiple es mejor para el caso de los datos de California. Con base en los porcentajes de resultados de las métricas de medición de los siguientes artículos [20], [64], [67], [66] podemos determinar las métricas no tienen anomalías y corresponden a un buen resultado de predicción.

6.5.2. Predicción del conjunto de prueba para granja Tennen-Alemania

Esta sección muestra los resultados obtenidos por el modelo ARIMA-múltiple, así como los resultados obtenidos por los modelos óptimos ARIMA para el conjunto de datos de Tennen-Alemania.

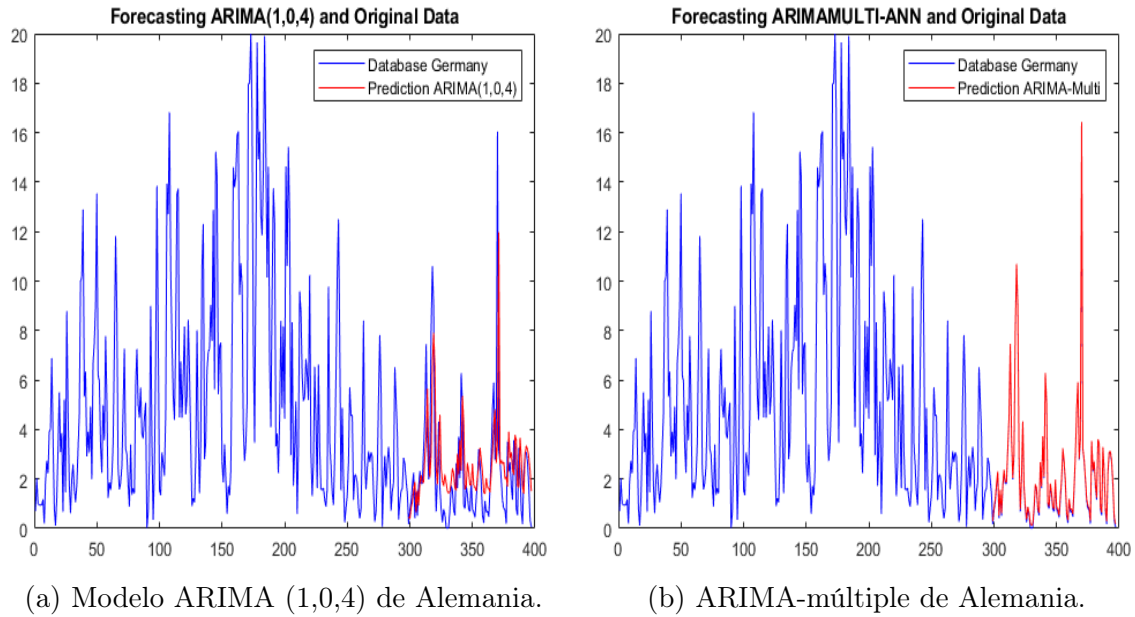


Figura 6.11: Resultado de predicción de los modelos con los datos de Alemania.

La figura 6.11a es el resultado de los datos de prueba evaluados en el modelo ARIMA-Alemania óptimo, mientras que la figura 6.11b es la respuesta de la salida del ARIMA-múltiple para Alemania. Como en el anterior modelo de California, la predicción en la figura 6.11b también coincide con una mejora de respuesta a comparación del ARIMA(1,0,4) óptimo para Tennen-Alemania. Sin embargo, la respuesta gráfica debe compararse de forma numérica en la siguiente tabla de mediciones de error.

Model	MAE	SMAPE	RMSE	RSQ
ARIMA (1,0,4) Alemania	1.4644	0.6931	2.1952	0.1435
ARIMA-Multi Alemania	0.0687	0.1239	0.0811	0.9988

Tabla 6.5: Métricas para los modelos de Alemania

La tabla 6.5 muestra una comparativa de las métricas para los datos de prueba aplicados en Alemania-Tennet. Cada una de las métricas en el modelo ARIMA-Múltiple presenta un error menor en promedio del 80 % respecto al otro modelo. En el caso del MAE presenta una mejora del 89 % y la métrica nos indica que cuanto más cercano a 0, se obtiene un mejor rendimiento. Para la métrica SMAPE la mejora es del 82.1 %, al igual que la métrica MAE nos indica que cuando se acerca a 0 el rendimiento del modelo es de buena predicción. Con la métrica RMSE tiene un comportamiento igual a las anteriores, la mejora que presenta el modelo múltiple con esta métrica es de 89 % respecto al modelo ARIMA. La última métrica RSQ tiene una mejora del 84.6 %, en este caso cuando la métrica es cercana 1, el modelo se acopla mejor a los datos originales. Se puede observar que ARIMA-múltiple de Alemania respecto a su ARIMA(1,0,4) óptimo, obtiene resultados graficos muy cercanos a los datos del conjunto original, en cuanto a las métricas de errores, estas muestran que el modelo ARIMA-múltiple es mejor para el caso de los datos de Alemania-Tennet. Con base en los porcentajes de resultados de las métricas de medición de los siguientes artículos [20], [67], [66], [64] podemos determinar las métricas no tienen anomalías y corresponden a un buen resultado de predicción.

6.5.3. Predicción del conjunto de prueba para granja 1 Kaggle

Esta sección muestra los resultados obtenidos por el modelo ARIMA-múltiple, así como los resultados obtenidos por el modelo óptimo ARIMA(1,1,2) para el conjunto de datos de Kaggle.

Para el caso de la base de datos Kaggle, se obtuvo un resultado no esperado al comparar el modelo ARIMA múltiple y el ARIMA óptimo.

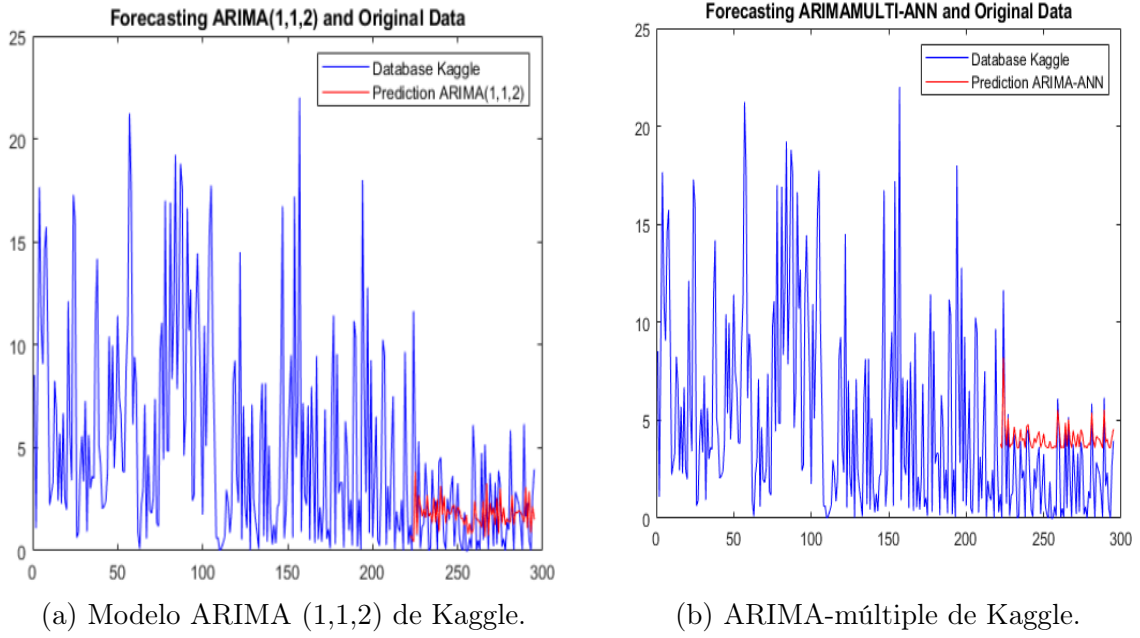


Figura 6.12: Resultado de predicción de los modelos con los datos de Kaggle.

La figura 6.12a es el resultado de datos de prueba evaluados en el modelo ARIMA-Kaggle óptimo, mientras que la figura 6.12b es la respuesta del modelo ARIMA múltiple para Kaggle. Contrario a los casos anteriores, Kaggle no logra obtener una buena predicción en su modelo múltiple a comparación del modelo ARIMA óptimo. La tabla 6.6 es la comparativa de métricas de ambos modelos, en este caso el modelo ARIMA óptimo de Kaggle es un 14.32 % con mejores resultados que el ARIMA-múltiple. Aunque las métricas de SMAPE con (5.72 %) y RMSE con (5.35 %) muestran que el ARIMA múltiple tiene mejor respuesta de predicción, la métrica RSQ califica a ambos modelos como insuficientes para predecir. La respuesta obtenida en las gráficas nos muestran que la predicción no es nada precisa. Para encontrar una solución se ha propuesto una técnica de compensación que mejora el modelo ARIMA-múltiple.

Model	MAE	SMAPE	RMSE	RSQ
ARIMA(1,1,2) Kaggle	1.6051	0.9938	2.2486	-0.1485
ARIMA-Multi Kaggle	1.8351	0.9369	2.1223	-0.0231

Tabla 6.6: Métricas para los modelos de Kaggle

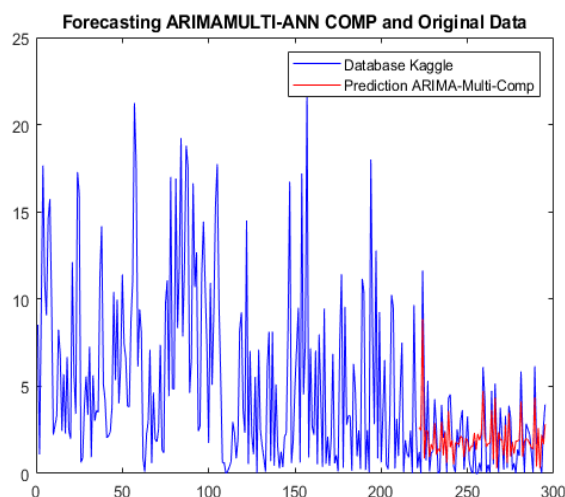


Figura 6.13: Modelo de ARIMA-múltiple compensado

La figura 6.13 es la respuesta del diagrama 5.6 del modelo ARIMA-múltiple compensado, el modelo múltiple se resta con los residuales del modelo ARIMA óptimo. Esto conlleva a un modelo compensado con el error y la respuesta nos presenta una mejora de su antecesor el ARIMA-múltiple Kaggle.

Model	MAE	SMAPE	RMSE	RSQ
ARIMA-Multi Kaggle compensado	1.2064	0.8582	1.435	0.5322
ARIMA-Multi Kaggle	1.8351	0.9369	2.1223	-0.0231
ARIMA(1,1,2) Kaggle	1.6051	0.9938	2.2486	-0.1485

Tabla 6.7: Métricas para los modelos Kaggle y el modelo compensado

La tabla 6.7 determina la mejora del modelo múltiple compensado para Kaggle a través de las métricas de error. En el caso del MAE presenta una mejora del 24% y la métrica nos indica que cuanto más cercano a 0 se obtiene un mejor rendimiento. Para la métrica SMAPE la mejora es del 13.64% del modelo compensado hacia el ARIMA óptimo para Kaggle. Con la métrica RMSE el modelo múltiple compensado es de 36.18% mejor respecto al modelo ARIMA. La última métrica es el RSQ con una mejora del 69.6% en el modelo ARIMA-múltiple compensado, en este caso cuando la métrica es cercana 1, el modelo se acopla mejor a los datos originales. Se puede observar que el modelo ARIMA-múltiple compensado de Kaggle respecto a su ARIMA óptimo y ARIMA-múltiple presenta una mejora en la predicción. En conclusión, no hace falta comparar directamente las métricas del modelo ARIMA-múltiple compensado contra el ARIMA-múltiple, puesto que las mejoras contra el modelo ARIMA óptimo de Kaggle son por encima del 10%, mostrando que están por encima de los resultados y con una respuesta gráfica superior a ambos modelos.

Capítulo 7

Conclusiones y trabajos futuros

7.1. Conclusiones

Se construyó un algoritmo de estrategia multimodelo con base en modelos ARIMA y redes neuronales. Generalmente en distintos artículos se menciona la unión de estos modelos por medio de las sumas de predicciones en las que separan la parte lineal, la parte no lineal y el error ($\hat{y} = y_l + y_{nl} + e_t$). Podríamos decir que es un modelo más de compensación que una unión de estos. Por ello el planteamiento que consideramos fue entrenar los modelos ARIMA hasta llegar a los mejores modelos que puede tener cada conjunto de datos. Utilizando el criterio AIC y basándonos en la teoría que proporciona la transferencia de aprendizaje, de entrenar una red neuronal a partir de modelos ya pre-entrenados. Desde mi punto de vista esto significaba otorgar una mayor cantidad de características a nuestros conjuntos, porque en la actualidad los modelos más modernos implican tomar en cuenta las condiciones meteorológicas que presenta el área de ubicación en la que está localizada la granja. Con el fin de determinar el comportamiento del viento a partir de temporadas. Sin embargo, se requiere una gran cantidad de datos, que lamentablemente en un país como México es complicado conseguir una base de datos con esa información y llega ser hasta cierto punto imposible, debido a que hay demasiada restricción por parte de las empresas que trabajan con este tipo de energías. Por ello el planteamiento fue obtener características de lugares completamente distintos y hacer la unión de modelos para que se consideren más características distintas áreas, el lugar de solo proponer más datos de granjas eólicas ubicadas en un mismo lugar. Cabe resaltar que nuestro objetivo fue un poco acotado debido a que no pudimos compararlo con otro modelo un poco más robusto, pero en un futuro se planea realizar una comparativa con otros modelos y una mejora para el modelo realizado.

Otro de los objetivos planteado fue utilizar un conjunto (Kaggle) con menos datos que los otros 2 conjuntos. Como bien hemos mencionado realmente pocos modelos de predicción pueden trabajar con conjuntos que contienen menos datos, en este caso era cerca de un 10% menos datos que los otros conjuntos. Los modelos ARIMA

no pueden realizar una buena predicción con conjuntos con menos datos y las ANN mucho menos. Se requiere de una gran cantidad de datos para obtener suficientes características y obtener un modelo que pronostique buenas predicciones. Nuevamente a través de la transferencia de aprendizaje, buscamos una oportunidad para mejorar el conjunto de datos, y de esta forma proporcionarle más características a un modelo de predicción; dando como resultado un modelo con mayor estabilidad y con mejoras de predicción. Aunque esto no resultó en una mejora para los 3 modelos planteados, dos de los tres conjuntos, lograron obtener aprendizaje de un conjunto mayor que dio mejoras en su modelo. Los datos de Kaggle crearon el único modelo donde no existe una mejora con solo aplicar la estrategia multimodelo. Por ello consideramos apoyarnos en la idea de hacer una compensación al modelo, sumando el modelo obtenido menos la predicción del modelo ARIMA para ese conjunto de datos. Al realizar las pruebas de la compensación nos otorgó mejores resultados que el modelo ARIMA óptimo para Kaggle y aunque no es un porcentaje muy alto de mejora, se obtiene un modelo estable con buena predicción. Aún falta fortalecer esta estrategia de compensación para un modelo con pocos datos, debido a que es posible que se requieran más características, y eso solo se consigue con más conjuntos de datos.

El último objetivo fue obtener resultados aceptables en las métricas comparativas, cómo bien podemos observar en la sección 6.5 al comparar ambos modelos los resultados obtenidos tienen mejoras altamente buenas en comparación con los modelos ARIMA. Aunque el objetivo está un poco acotado como tal, debido a que la comparativa es con el modelo ARIMA y no con otros modelos actuales, el modelo es bastante estable. Las últimas pruebas que realicé fue ver si nuestro modelo logra mantener buenas predicciones cuando se somete a una prueba de repeticiones, Se sometió de 50-100 repeticiones, entregando un cambio en las métricas del 1 % en promedio para el caso de 100 repeticiones. Lo que nos demuestra que no fue un caso de la casualidad el que obtenga buenas predicciones. Regresando al punto de la comparativa, como tal la idea es continuar con este modelo y llevarlo a comparar en un primer punto con otros mecanismos de aprendizaje automático; cambiar la red neuronal por otro mecanismo como SVM, KNN u otro mecanismo. El segundo punto es comparar el modelo ARIMA-múltiple con modelos más robustos para verificar que las métricas vuelven a estar a favor de nuestro modelo.

7.2. Trabajo a futuro

El trabajo tiene varias áreas de oportunidad en las que se puede aprovechar el mismo modelo o complementar para obtener otros beneficios mejores, a continuación, listaré varias propuestas:

- La primera propuesta es implementar una fase de descomposición de frecuencia en las series de tiempo antes de entrar al modelo ARIMA, para obtener más características tanto de frecuencia alta y baja. Al tener más series de tiempo por usar el método de descomposición de frecuencias le otorgará a la red neuronal características más provechosas, obteniendo un modelo más completo.
- La siguiente propuesta es ocupar distintos métodos de aprendizaje automático y verificar cual es el mejor para la unión de los modelos. En palabras simples es cambiar la ANN por un SVM, KNN, u otros modelos. Esto para observar el comportamiento y los resultados que nos arrojan las métricas de medición y con base en ello determinar si podemos encontrar un método mejor o seguir con una red neuronal.
- Conseguir los datos fue complicado, pero una propuesta interesante sería conseguir datos de granjas eólicas ubicadas en México y poner a prueba el modelo. Durante el proceso de la tesis, intentamos hablar con personas que ya habían desarrollado artículos respecto a energía eólica ocupando datos de CFE. Sin embargo, no fue posible que nos prestaran los datos.
- Probar el funcionamiento del modelo con distintas series de tiempo que tengan una variable distinta a la energía, en particular el modelo debe funcionar para distintas variables (predicción de movimientos sísmicos, por ejemplo) que correspondan a una serie de tiempo.
- Obtener una base de datos con datos meteorológicos y con base en ello crear series de tiempo para construir modelos más precisos. Realmente en muchos artículos mencionan que para tener una predicción más completa se deben analizar los datos meteorológicos de la granja eólica, pero conseguir ese tipo de datos es demasiado complicado. Por ello una buena prueba sería conseguir ese tipo de datos y comprobar el funcionamiento del modelo.

Bibliografía

- [1] Harsh S Dhiman, Dipankar Deb, and Valentina Emilia Balas. *Supervised machine learning in wind forecasting and ramp event prediction*. Academic Press, 2020.
- [2] Wen-chuan Wang, Kwok-wing Chau, Dong-mei Xu, and Xiao-Yun Chen. Improving forecasting accuracy of annual runoff time series using arima based on eemd decomposition. *Water Resources Management*, 29(8):2655–2675, 2015.
- [3] Clare Breidenich, Daniel Magraw, Anne Rowley, and James W Rubin. The kyoto protocol to the united nations framework convention on climate change. *American Journal of International Law*, 92(2):315–331, 1998.
- [4] Barbara G Brown, Richard W Katz, and Allan H Murphy. Time series models to simulate and forecast wind speed and wind power. *Journal of Applied Meteorology and Climatology*, 23(8):1184–1195, 1984.
- [5] Global Wind Energy Council. Gwec| global wind report 2021. 2021.
- [6] Roberto Cadenas Tovar, Gaffie Saldivar Urquiza, et al. Central eoloeléctrica la venta ii. 2007.
- [7] Erasmo Cadenas, Wilfrido Rivera, Rafael Campos-Amezcuca, and Christopher Heard. Wind speed prediction using a univariate arima model and a multivariate narx model. *Energies*, 9(2):109, 2016.
- [8] Rajneesh Sharma, Tushar Shikhola, and Jaspreet Kaur Kohli. Modified fuzzy q-learning based wind speed prediction. *Journal of Wind Engineering and Industrial Aerodynamics*, 206:104361, 2020.
- [9] Manassakan Sanayha and Peerapon Vateekul. Model-based deep reinforcement learning for wind energy bidding. *International Journal of Electrical Power and Energy Systems*, 136:107625, 2022.
- [10] Fang Liu, Ranran Li, and Aliona Dreglea. Wind speed and power ultra short-term robust forecasting based on takagi–sugeno fuzzy model. *Energies*, 12(18), 2019.

- [11] A.Z. Dhunny, J.R.S. Doorga, Z. Allam, M.R. Lollchund, and R. Boojhawon. Identification of optimal wind, solar and hybrid wind-solar farming sites using fuzzy logic modelling. *Energy*, 188:116056, 2019.
- [12] Koffi Agbeblewu Dotche, Adekunla Akim Salami, Koffi Mawugno Kodjo, Hadnane Ouro-Agbake, and Koffi-Sa Bedja. Wind speed prediction based on support vector regression method: a case study of lome-site. In *2019 IEEE PES/IAS PowerAfrica*, pages 267–272, 2019.
- [13] Carlos Ruiz, Carlos M. Alaz, and Jose R. Dorronsoro. Multitask support vector regression for solar and wind energy prediction. *Energies*, 13(23), 2020.
- [14] Pawar A., Jape V.S., and Mathew S. Wind power forecasting using support vector machine model in rstudio. In *Cognitive Informatics and Soft Computing*, volume 768, pages 289–298, 2019.
- [15] Noman Shabbir, Roya AhmadiAhangar, Lauri Katt, Muhamamd N. Iqbal, and Argo Rosin. Forecasting short term wind energy generation using machine learning. In *2019 IEEE 60th International Scientific Conference on Power and Electrical Engineering of Riga Technical University (RTU CON)*, pages 1–4, 2019.
- [16] Sumanta Pasari, Aditya Shah, and Utkarsh Sirpurkar. Wind energy prediction using artificial neural networks. In *Enhancing Future Skills and Entrepreneurship*, pages 101–107. Springer International Publishing, 2020.
- [17] R Kaja Bantha Navas, S Prakash, and T Sasipraba. Artificial neural network based computing model for wind speed prediction: A case study of coimbatore, tamil nadu, india. *Physica A: Statistical Mechanics and its Applications*, 542:123–383, 2020.
- [18] Amir Ahadi and Xiaodong Liang. Wind speed time series predicted by neural network. In *2018 IEEE Canadian Conference on Electrical Computer Engineering (CCECE)*, pages 1–4, 2018.
- [19] M. Madhiarasan. Accurate prediction of different forecast horizons wind speed using a recursive radial basis function neural network. *Protection and Control of Modern Power Systems*, 5:22, 2020.
- [20] Erasmo Cadenas and Wilfrido Rivera. Wind speed forecasting in three different regions of mexico, using a hybrid arima–ann model. *Renewable Energy*, 35(12):2732–2738, 2010.
- [21] G Peter Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.
- [22] Pramod Jain. *Wind energy engineering*. McGraw-Hill Education, 2016.

- [23] Yun Wang, Qinghua Hu, Dipti Srinivasan, and Zheng Wang. Wind power curve modeling and wind power forecasting with inconsistent data. *IEEE Transactions on Sustainable Energy*, 10(1):16–25, 2018.
- [24] GH Riahy and M Abedi. Short term wind speed forecasting for wind turbine applications using linear prediction method. *Renewable energy*, 33(1):35–41, 2008.
- [25] Jinhua Zhang, Jie Yan, David Infield, Yongqian Liu, and Fue-sang Lien. Short-term forecasting and uncertainty analysis of wind turbine power based on long short-term memory network and gaussian mixture model. *Applied Energy*, 241:229–244, 2019.
- [26] Miguel A Prósper, Carlos Otero-Casal, Felipe Canoura Fernández, and Gonzalo Miguez-Macho. Wind power forecasting for a real onshore wind farm on complex terrain using wrf high resolution simulations. *Renewable energy*, 135:674–686, 2019.
- [27] Jie Yan, Hao Zhang, Yongqian Liu, Shuang Han, Li Li, and Zongxiang Lu. Forecasting the high penetration of wind power on multiple scales using multi-to-multi mapping. *IEEE Transactions on Power Systems*, 33(3):3276–3284, 2017.
- [28] Guillermo Santamaría-Bonfil, A Reyes-Ballesteros, and CJRE Gershenson. Wind speed forecasting for wind farms: A method based on support vector regression. *Renewable Energy*, 85:790–809, 2016.
- [29] Yun Wang, Runmin Zou, Fang Liu, Lingjun Zhang, and Qianyi Liu. A review of wind speed and wind power forecasting with deep neural networks. *Applied Energy*, 304:117766, 2021.
- [30] Hui Liu, Chao Chen, Xinwei Lv, Xing Wu, and Min Liu. Deterministic wind energy forecasting: A review of intelligent predictors and auxiliary methods. *Energy Conversion and Management*, 195:328–345, 2019.
- [31] HZ Wang, GB Wang, GQ Li, JC Peng, and YT Liu. Deep belief network based deterministic and probabilistic wind speed forecasting approach. *Applied Energy*, 182:80–93, 2016.
- [32] Xinyu Zhao, Na Jiang, Jinfu Liu, Daren Yu, and Juntao Chang. Short-term average wind speed and turbulent standard deviation forecasts based on one-dimensional convolutional neural network and the integrate method for probabilistic framework. *Energy Conversion and Management*, 203:112239, 2020.
- [33] Xiaohui Yuan, Chen Chen, Min Jiang, and Yanbin Yuan. Prediction interval of wind power using parameter optimized beta distribution based lstm model. *Applied Soft Computing*, 82:105550, 2019.

- [34] Jianming Hu, Jiani Heng, Jiemei Wen, and Weigang Zhao. Deterministic and probabilistic wind speed forecasting with de-noising-reconstruction strategy and quantile regression based algorithm. *Renewable Energy*, 162:1208–1226, 2020.
- [35] Yixiao Yu, Xueshan Han, Ming Yang, and Jiajun Yang. Probabilistic prediction of regional wind power based on spatiotemporal quantile regression. *IEEE Transactions on Industry Applications*, 56(6):6117–6127, 2020.
- [36] Abhishek Banik, Chinmaya Behera, Tirunagaru V Sarathkumar, and Arup Kumar Goswami. Uncertain wind power forecasting using lstm-based prediction interval. *IET Renewable Power Generation*, 14(14):2657–2667, 2020.
- [37] Ma Lei, Luan Shiyan, Jiang Chuanwen, Liu Hongling, and Zhang Yan. A review on the forecasting of wind speed and generated power. *Renewable and sustainable energy reviews*, 13(4):915–920, 2009.
- [38] 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.
- [39] Yun Wang, Yifen Li, Runmin Zou, Aoife M Foley, Dlzar Al Kez, Dongran Song, Qinghua Hu, and Dipti Srinivasan. Sparse heteroscedastic multiple spline regression models for wind turbine power curve modeling. *IEEE Transactions on Sustainable Energy*, 12(1):191–201, 2020.
- [40] Qinkai Han, Fanman Meng, Tao Hu, and Fulei Chu. Non-parametric hybrid models for wind speed forecasting. *Energy Conversion and Management*, 148:554–568, 2017.
- [41] Kalid Yunus, Torbjörn Thiringer, and Peiyuan Chen. Arima-based frequency-decomposed modeling of wind speed time series. *IEEE Transactions on Power Systems*, 31(4):2546–2556, 2015.
- [42] Rajesh G Kavasseri and Krithika Seetharaman. Day-ahead wind speed forecasting using f-arima models. *Renewable Energy*, 34(5):1388–1393, 2009.
- [43] Othman Ait Maatallah, Ajit Achuthan, Kerop Janoyan, and Pier Marzocca. Recursive wind speed forecasting based on hammerstein auto-regressive model. *Applied Energy*, 145:191–197, 2015.
- [44] Jianzhou Wang, Shiqi Wang, and Wendong Yang. A novel non-linear combination system for short-term wind speed forecast. *Renewable Energy*, 143:1172–1192, 2019.
- [45] Yun Wang, Jianzhou Wang, and Xiang Wei. A hybrid wind speed forecasting model based on phase space reconstruction theory and markov model: A case study of wind farms in northwest china. *Energy*, 91:556–572, 2015.

- [46] Hui Liu, Xiwei Mi, and Yanfei Li. Smart multi-step deep learning model for wind speed forecasting based on variational mode decomposition, singular spectrum analysis, lstm network and elm. *Energy Conversion and Management*, 159:54–64, 2018.
- [47] Wenlong Fu, Kai Wang, Jiawen Tan, and Kai Zhang. A composite framework coupling multiple feature selection, compound prediction models and novel hybrid swarm optimizer-based synchronization optimization strategy for multi-step ahead short-term wind speed forecasting. *Energy Conversion and Management*, 205:112461, 2020.
- [48] Carlos A Severiano, Petrônio Cândido de Lima e Silva, Miri Weiss Cohen, and Frederico Gadelha Guimarães. Evolving fuzzy time series for spatio-temporal forecasting in renewable energy systems. *Renewable Energy*, 171:764–783, 2021.
- [49] Federico Cassola and Massimiliano Burlando. Wind speed and wind energy forecast through kalman filtering of numerical weather prediction model output. *Applied energy*, 99:154–166, 2012.
- [50] Jingjing Song, Jianzhou Wang, and Haiyan Lu. A novel combined model based on advanced optimization algorithm for short-term wind speed forecasting. *Applied Energy*, 215:643–658, 2018.
- [51] Xiaohui He, Ying Nie, Hengliang Guo, and Jianzhou Wang. Research on a novel combination system on the basis of deep learning and swarm intelligence optimization algorithm for wind speed forecasting. *IEEE Access*, 8:51482–51499, 2020.
- [52] Lianqing Song, Qingyun Xie, Yongkang He, and Pengju Dang. Ultra-short-term wind power combination forecasting model based on meemd-sae-elman. In *2020 IEEE 4th Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, volume 1, pages 1844–1850. IEEE, 2020.
- [53] Mehdi Neshat, Meysam Majidi Nezhad, Ehsan Abbasnejad, Seyedali Mirjalili, Lina Bertling Tjernberg, Davide Astiaso Garcia, Bradley Alexander, and Markus Wagner. A deep learning-based evolutionary model for short-term wind speed forecasting: A case study of the lillgrund offshore wind farm. *Energy Conversion and Management*, 236:114002, 2021.
- [54] Zhewen Niu, Zeyuan Yu, Wenhua Tang, Qinghua Wu, and Marek Reformat. Wind power forecasting using attention-based gated recurrent unit network. *Energy*, 196:117081, 2020.
- [55] Min Ding, Hao Zhou, Hua Xie, Min Wu, Yosuke Nakanishi, and Ryuichi Yokoyama. A gated recurrent unit neural networks based wind speed error correction model for short-term wind power forecasting. *Neurocomputing*, 365:54–61, 2019.

- [56] Jie Wan, Jinfu Liu, Guorui Ren, Yufeng Guo, Daren Yu, and Qinghua Hu. Day-ahead prediction of wind speed with deep feature learning. *International Journal of Pattern Recognition and Artificial Intelligence*, 30(05):1650011, 2016.
- [57] Hatem Mezaache and Hassen Bouzgou. Auto-encoder with neural networks for wind speed forecasting. In *2018 International Conference on Communications and Electrical Engineering (ICCEE)*, pages 1–5. IEEE, 2018.
- [58] Jing Shi, Jinmei Guo, and Songtao Zheng. Evaluation of hybrid forecasting approaches for wind speed and power generation time series. *Renewable and Sustainable Energy Reviews*, 16(5):3471–3480, 2012.
- [59] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.
- [60] Rizwan Mushtaq. Augmented dickey fuller test. 2011.
- [61] Pang-Ning Tan, Michael Steinbach, and Vipin Kumar. *Introduction to data mining*. Pearson Education India, 2016.
- [62] Davide Chicco, Matthijs J Warrens, and Giuseppe Jurman. The coefficient of determination r-squared is more informative than smape, mae, mape, mse and rmse in regression analysis evaluation. *PeerJ Computer Science*, 7:e623, 2021.
- [63] Hui Liu, Hong-qi Tian, and Yan-fei Li. Comparison of two new arima-ann and arima-kalman hybrid methods for wind speed prediction. *Applied Energy*, 98:415–424, 2012.
- [64] Hui Liu, Hong-qi Tian, and Yan-fei Li. An emd-recursive arima method to predict wind speed for railway strong wind warning system. *Journal of Wind Engineering and Industrial Aerodynamics*, 141:27–38, 2015.
- [65] Jianzhou Wang and Jianming Hu. A robust combination approach for short-term wind speed forecasting and analysis—combination of the arima (autoregressive integrated moving average), elm (extreme learning machine), svm (support vector machine) and lssvm (least square svm) forecasts using a gpr (gaussian process regression) model. *Energy*, 93:41–56, 2015.
- [66] Osamah Basheer Shukur and Muhammad Hisyam Lee. Daily wind speed forecasting through hybrid kf-ann model based on arima. *Renewable Energy*, 76:637–647, 2015.
- [67] Krishnaveny R Nair, V Vanitha, and M Jisma. Forecasting of wind speed using ann, arima and hybrid models. In *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, pages 170–175. IEEE, 2017.

- [68] Cristian-Dragos Dumitru and Adrian Gligor. Wind energy forecasting: a comparative study between a stochastic model (arima) and a model based on neural network (ffann). *Procedia manufacturing*, 32:410–417, 2019.
- [69] Fei Zhang, Peng-Cheng Li, Lu Gao, Yong-Qian Liu, and Xiao-Ying Ren. Application of autoregressive dynamic adaptive (arda) model in real-time wind power forecasting. *Renewable Energy*, 169:129–143, 2021.
- [70] Mehdi Khashei and Mehdi Bijari. An artificial neural network (p, d, q) model for timeseries forecasting. *Expert Systems with applications*, 37(1):479–489, 2010.
- [71] Li Wang, Hao-fei Zou, Jia Su, Ling Li, and Sohail Chaudhry. An arima-ann hybrid model for time series forecasting. *Systems Research and Behavioral Science*, 30(3):244–259, 2013.
- [72] Ümit Çavuş Büyüksahin and Şeyda Ertekin. Improving forecasting accuracy of time series data using a new arima-ann hybrid method and empirical mode decomposition. *Neurocomputing*, 361:151–163, 2019.
- [73] Kaggle-global energy forecasting competition 2012, 2012.
- [74] California ISO-renewables and emissions reports, 2013.
- [75] netztransparenz-informationsplattform der deutschen Übertragungsnetzbetreiber, 2020.