



UNIVERSIDAD DE CHILE  
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS  
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

# **USO DE SERIES DIFUSAS Y VARIABLES ESTOCÁSTICAS PARA EL DESARROLLO DE MODELOS PREDICTIVOS DE LA DEMANDA DE ENERGÍA ELÉCTRICA**

MEMORIA PARA OPTAR AL TÍTULO DE INGENIERO CIVIL ELÉCTRICO

JOSÉ RUBIO CIENFUEGOS

PROFESOR GUÍA:  
José Miguel Rubio León

MIEMBROS DE LA COMISIÓN:  
PROFESOR 2  
PROFESOR 3

Este trabajo ha sido parcialmente financiado por:  
NOMBRE INSTITUCIÓN

SANTIAGO DE CHILE  
2021

# USO DE SERIES DIFUSAS Y VARIABLES ESTOCÁSTICAS PARA EL DESARROLLO DE MODELOS PREDICTIVOS DE LA DEMANDA DE ENERGÍA ELÉCTRICA

El siguiente trabajo tiene como finalidad exponer los beneficios de implementar modelos de series difusas para la predicción de la demanda de la energía eléctrica. Para ello se utiliza una base de datos pública que contiene información sobre el consumo de energía de España durante 4 años.<sup>1</sup> Hay que destacar que además de información sobre el consumo de energía, esta base también tiene datos sobre datos relacionados al consumo, como la generación de diversas fuentes de energía y el precio del mercado eléctrico.

Si bien la finalidad principal es utilizar la lógica difusa para la realización de modelos de predicción de la demanda de la energía eléctrica. Para poder analizar su eficacia se debe comparar el rendimiento de este sistema con otros algoritmos que realizan este tipo de trabajo, para ello también se programaron modelos de *machine learning*, para poder comparar los rendimientos de los sistemas difusos presentados en este trabajo.

Para poder realizar las predicciones por los modelos mencionados anteriormente, se tuvo que convertir los datos de la demanda energética en una serie temporal. Tras este punto, para aplicar los modelos de series difusas en primer lugar esta información se tuvo que llevar al dominio de la lógica difusa, lo que se conoce como **universo de discurso**, donde las funciones que realizan este proceso se denominan **particiones**, donde crean pequeños subconjuntos, donde los datos procesados se les atribuye un grado de pertenencia dependiendo de su valor. Una vez que la información se encuentra como su universo de discurso, los modelos aprenden como se comporta cada área (subconjunto) y extraer reglas de estas a través de los patrones de las series temporales, donde estas reglas indican como se comportan las particiones consigo mismas. El proceso de llevar la información al dominio difuso se llama **fuzzyficación**, y una vez que se ha aplicado la predicción a los datos deseados, entonces se debe llevar al dominio original (tiempo) de la información, es decir se debe realizar una **defuzzyficación**. Por otro lado, el uso de los modelos de *machine learning* no requirió los procesos mencionados anteriormente, simplemente se procesaron las series temporales en el mismo dominio convirtiendo la data en diversas series temporales que representaron diferentes períodos de consumo.

Para realizar el entrenamiento de los modelos mencionados se utilizaron librerías de *machine learning* y de series difusas en python, donde los resultados obtenidos indican el porcentaje de precisión promedio del conjunto evaluado. Por ende la finalidad de este trabajo será encontrar el modelo difuso junto con sus hiperparámetros que pudieron desarrollar de mejor manera la predicción de la demanda de la energía eléctrica.

---

<sup>1</sup> Los detalles de esta base de datos más su fuente se mencionan más adelante.

# Tabla de Contenido

<b>1. Marco Teórico</b>	<b>3</b>
1.1. Series Temporales. . . . .	3
1.2. Series Temporales difusas. . . . .	4
1.2.1. Lógica difusa . . . . .	4
1.2.2. Serie Temporal Difusa . . . . .	5
1.2.3. Operaciones difusas . . . . .	5
1.2.3.1. Fuzzyficación. . . . .	5
1.2.3.2. Intersección difusa. . . . .	5
1.2.3.3. Reglas difusas e inferencia. . . . .	5
1.3. Modelos de predicción . . . . .	6
1.3.1. Modelos de machine learning. . . . .	6
1.3.1.1. Redes Neuronales Artificiales. . . . .	6
1.3.1.2. Random Forest. . . . .	8
1.3.2. Modelos de series temporales difusas. . . . .	9
1.3.2.1. Parámetros relevantes . . . . .	9
1.3.2.2. Modelos monovariantes y multivariantes. . . . .	11
1.3.2.3. Proceso de un modelo difuso. . . . .	12
1.4. Métricas de error . . . . .	13
1.4.1. RMSE . . . . .	13
1.4.2. MAPE. . . . .	13
<b>2. Base de datos.</b>	<b>14</b>
2.1. Dataset. . . . .	14
2.2. Análisis exploratorio de la demanda disponible. . . . .	15
<b>3. Metodología.</b>	<b>16</b>
3.1. Preprocesamiento de datos. . . . .	17
3.2. Configuración de modelos. . . . .	18
3.2.1. Modelos de machine learning. . . . .	18
3.2.2. Modelos de series difusas. . . . .	18
3.2.2.1. Partición. . . . .	18
3.2.2.2. Orden. . . . .	19
3.2.2.3. Tipos de modelos. . . . .	19
3.3. Entrenamiento . . . . .	20
<b>4. Resultados.</b>	<b>22</b>
4.1. Modelos de ML. . . . .	22
4.1.1. Red Neuronal . . . . .	22

4.1.2.	Random Forest . . . . .	23
4.1.3.	Rendimiento de modelos de ML . . . . .	23
4.2.	Modelos de series difusas. . . . .	24
4.2.1.	Modelos monovariantes. . . . .	24
4.2.1.1.	HOFTS . . . . .	24
4.2.1.2.	WHOFTS . . . . .	24
4.2.1.3.	PWFTS . . . . .	25
4.2.1.4.	Rendimiento de modelos de una variable. . . . .	25
4.2.2.	Modelos multivariantes. . . . .	26
4.2.2.1.	MVFTS . . . . .	26
4.2.2.2.	Weighted MVFTS . . . . .	26
4.2.2.3.	FIG - FTS . . . . .	27
4.2.2.4.	Rendimiento de modelos multivariantes. . . . .	27
4.2.3.	Rendimiento General. . . . .	27
4.3.	Análisis . . . . .	29
<b>5.</b>	<b>Conclusiones.</b>	<b>30</b>
	<b>Bibliografía</b>	<b>31</b>

# Índice de Tablas

2.1.	Información resumida disponible en la base de datos utilizada, debido a la gran cantidad de columnas, solo se mostraron estos datos. . . . .	14
2.2.	Datos de interés de la demanda disponible. . . . .	15
3.1.	Demanda registrada por el TSO español, junto con el pronóstico de esta y su fecha . .	17
3.2.	Series temporales creadas a partir de los datos de la demanda junto con la fecha de registro. . . . .	17
3.3.	Hiperparámetros de la Red Neuronal utilizada. . . . .	18
3.4.	Hiperparámetros del sistema Random Forest utilizado. . . . .	18
3.5.	Características generales de los modelos monovariantes usados, las ponderaciones mencionadas hacen referencia al cálculo de importancia de los conjuntos difusos generados.	20
3.6.	Configuraciones generales de los modelos multivariantes, la partición inicial fue la aplicada a la variable objetivo. . . . .	20
3.7.	Cantidad de muestras de conjuntos por modelo . . . . .	20
4.1.	Rendimiento de modelos de machine learning supervisados junto con el MAPE obtenido por el TSO . . . . .	23
4.2.	Error Porcentual Medio Absoluto obtenido por los modelos difusos monovariantes. . .	25
4.3.	Error Porcentual Medio Absoluto obtenido por los modelos difusos multivariantes. . .	27
4.4.	Modelos difusos probados junto con su rendimiento . . . . .	28

# Índice de Ilustraciones

1.1.	Serie temporal del Índice Mensual de Actividad Económica (IMACEC) del sector minero, no minero y en conjunto entre 2019 - 2039. . . . .	3
1.2.	Diferencias entre la asociación de la altura a los conjuntos de alto y bajo mediante lógica clásica (a) y lógica difusa (b) . . . . .	4
1.3.	Diagrama de neurona artificial (a), y agrupación de estas en capas de una pequeña red (b) . . . . .	7
1.4.	Diagrama del proceso de un algoritmo random forest. . . . .	8
1.5.	Comportamiento de predicciones de una función seno realizados por modelos difusos de diferentes ordenes. . . . .	9
1.6.	Partición Grid aplicado a 10 conjuntos difusos. . . . .	10
1.7.	Partición tipo CMeans aplicado a 10 conjuntos difusos . . . . .	10
1.8.	Predicciones de un modelo difuso para una función seno con distintas cantidades de particiones . . . . .	10
1.9.	Partición Grid con función de membresía triangular aplicado a 20 conjuntos difusos. .	11
1.10.	Partición Grid con función de membresía trapecio aplicado a 20 conjuntos difusos. .	11
1.11.	Partición Grid con función de membresía gaussiana aplicado a 20 conjuntos difusos. .	11
2.1.	Visualización gráfica de la demanda total y los valores de esta durante los primeros 10 días de 2015. . . . .	15
3.1.	Visualización de la partición Grid configurada con 35 niveles y 3 tipos de funciones de membresía. El eje y representa el grado de pertenencia de algún valor temporal (eje x). .	19
3.2.	Partición temporal aplicado a las fechas registradas. . . . .	19
4.1.	Predicciones de la demanda eléctrica de la red neuronal descrita en la sección 3.2.1 . .	22
4.2.	Predicciones de la demanda eléctrica de la Random Forest descrito en la sección 3.2.1	23
4.3.	Comparación de las predicciones de los modelos junto con la predicción realizada por el TSO español . . . . .	23
4.4.	Predicciones de la demanda realizado por el modelo monovariable HOFTS para 3 ordenes . . . . .	24
4.5.	Predicciones de la demanda realizado por el modelo monovariable WHOFTS para 3 ordenes . . . . .	24
4.6.	Predicciones de la demanda realizado por el modelo monovariable PWFTS para 3 ordenes . . . . .	25
4.7.	Predicciones de la demanda realizado por el modelo multivariable MVFTS . . . . .	26
4.8.	Predicciones de la demanda realizado por el modelo multivariable WeightedMVFTS .	26
4.9.	Predicciones de la demanda realizado por el modelo multivariable FIG-FTS, hasta 3 vecinos en el uso de KNN. . . . .	27

# Introducción

La energía eléctrica es uno de los grandes motores de la humanidad desde finales del siglo XVIII, dado esto comprender todo lo relacionado a ella se convierte en algo primordial. En la actualidad no existe alguna industria que no requiera el uso de energía eléctrica, lo que genera una gran necesidad de administrar la energía generada, esto con la finalidad de mantener y desarrollar todas las actividades humanas que requieran el uso de electricidad. Debido a la gran importancia de la energía eléctrica, surgen los mercados eléctricos, donde estos analizan económicamente todo lo relacionado a la generación, transmisión y distribución de la energía, cuya finalidad es poder satisfacer todas las necesidades energéticas que requiera cada sector e industria de cada país, donde una de las variables más importantes para ver el comportamiento de este corresponde a la **demandas de energía eléctrica**, donde esta información indica cuanto es la energía requerida para que una entidad o serie de consumidores puedan abastecer sus necesidades.

Dado que las actividades humanas están sujetas a una gran cantidad de factores externos, la demanda de la energía no esta exenta de esto. El análisis y comprensión de esta variable es importante para el desarrollo del sector energético, especialmente para el ámbito de la generación de energía, donde esta última actividad aun se sigue realizando en su mayoría a partir de recursos no renovables (Petróleo: 32.89 %, Carbón: 29.16 %, Gas Natural: 23.40 % son las 3 fuentes de energía mas utilizada en el mundo<sup>[1]</sup>). Esta característica muestra que es necesario desarrollar modelos que permitan pronosticar la demanda, ya que de esta forma permitiría administrar de mejor manera la generación y consumo energético.

Al analizar la demanda como serie temporal, es decir viendo el comportamiento de la demanda alrededor del tiempo, se pueden encontrar comportamientos interesantes. Si se analizan los valores de demanda energéticas anuales, se aprecia un crecimiento de esta a través de los años, siendo esto la consecuencia del desarrollo de las industrias, aumento de la población, aumento de la tecnología, desarrollo económico, etc. Por otro lado, si se analiza la demanda con valores registrados por hora, entonces se aprecia un aumento de estas durante el día y una reducción durante la noche, donde esto indica una relación estrecha de esta variable con el desarrollo de las actividades laborales y cotidianas. Sin embargo, al analizar la demanda en estos intervalos de tiempo, en diversas ocasiones se tiene registros que no obedecen este comportamiento, e incluso valores que superan el máximo que se registraron en días anteriores. Como se mencionó anteriormente la demanda depende en gran medida del desarrollo de las actividades humanas, donde estas están sujetas a una gran cantidad de factores los cuales en muchos casos poseen comportamientos estocásticos (fallas en los sistemas eléctricos, eventos de gran convocatoria, evolución económico de la industria, clima por dar algunos ejemplos). Esto muestra que la demanda en una gran cantidad de ocasiones posee un comportamiento aleatorio, lo que dificulta su análisis al desarrollar modelos de predicción.

Debido a estas características y tambien a la importancia de la demanda, se han desarrollado

muchos sistemas que predicen esta variable, utilizando técnicas de todo tipo, utilizando *machine learning supervisado*, *deep learning*, sistemas autorregresivos como ARIMA, entre otros. Dentro de todos los modelos que realizan regresiones, el **uso de series difusas** han sido sistemas que han tenido mejores rendimientos en este campo, donde estos modelos además de ser muy variados permite incorporar de manera sencilla la información de variables relacionadas, ya que dentro de su proceso de entrenamiento se encuentra el análisis profundo de las relaciones entre las variables procesadas.

Dado lo anterior, el objetivo principal de este trabajo es explotar el potencial de las series temporales difusas para la predicción de la demanda de la energía eléctrica. Incorporando información relacionada a la demanda como el horario, el clima, o el precio de la energía alrededor del tiempo. Esto con la finalidad de desarrollar un modelo que pueda predecir los siguientes valores de la demanda con una mayor precisión. Para ello en las siguientes secciones de este trabajo se presentará lo siguiente:

- Una revisión sobre los conceptos más importantes para la comprensión del trabajo desarrollado en el Marco Teórico.
- Análisis profundo sobre la base de datos, mostrando las peculiaridades de la información disponible, especialmente de la demanda y los datos más relacionados a esta.
- La metodología utilizada para el desarrollo de los modelos destacando el preprocesamiento y el entrenamiento realizado.
- Resultados obtenidos tras la aplicación de 35 modelos con configuraciones diferentes junto con el análisis de los rendimientos obtenidos.
- Conclusiones sobre los resultados obtenidos, más puntos a desarrollar en un trabajo futuro para mejorar los resultados obtenidos.



# Capítulo 1

## Marco Teórico

Para una buena comprensión del trabajo desarrollado, se debe tener en claro los siguientes conceptos:

### 1.1. Series Temporales.

Es una sucesión de datos ordenados cronológicamente en intervalos de tiempo definidos, los datos pueden estar espaciados equitativamente (registro de la generación solar diaria de una planta fotovoltaica), o distintos (cantidad de terremotos anuales de una zona definida). Este tipo de representación ofrece muchas ventajas, ya que su análisis permite encontrar relaciones subyacentes en los datos (pueden ser de varias series temporales o consigo misma) donde estas pueden ser usadas para extrapolar el comportamiento hacia el pasado, periodos de pérdida de datos y futuro. Poseen 2 características relevantes: la **estacionalidad** que muestra si esta posee algún patrón alrededor del tiempo, y la **tendencia** que indica la evolución de esta variable (si los valores crecen, decrecen o se mantienen en el mismo rango).

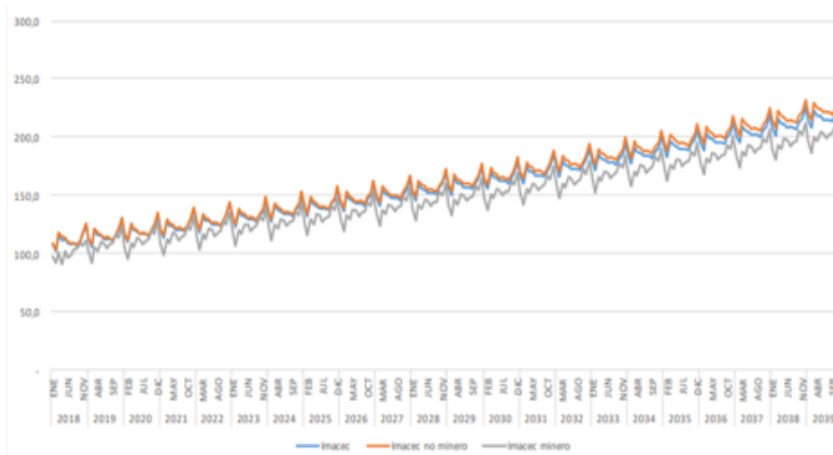


Figura 1.1: Serie temporal del Índice Mensual de Actividad Económica (IMACEC) del sector minero, no minero y en conjunto entre 2019 - 2039.

La figura 1.1 muestra la representación temporal de un indicador económico, en donde se puede apreciar una estacionalidad de alrededor de un año y una tendencia que muestra el crecimiento de estos valores.

## 1.2. Series Temporales difusas.

### 1.2.1. Lógica difusa

Este un tipo de lógica paraconsistente (un sistema de lógica que trata de las contradicciones de una forma atenuada), cuya principal diferencia de la lógica clásica es que no toma todas las expresiones como completamente ciertas o falsas.

Este lógica permite abordar los problemas de decisiones de una forma muy interesante, ya que en el mundo real no se puede abstraer todo a un sistema binario. Por ejemplo: considere que se tiene un registro de la altura de las personas en donde esta información es utilizada para saber quien es alto o pequeño de este conjunto de datos (figura 1.2), donde para hacer la diferencia entre estos conjuntos se toma el valor 1.70 mts. Al utilizar la lógica normal se tiene que todos los datos estarán en alguno de estos conjuntos, sin embargo ¿sería correcto decir que alguien con estatura 1.69 mts es pequeño o 1.71 mts es alto?.

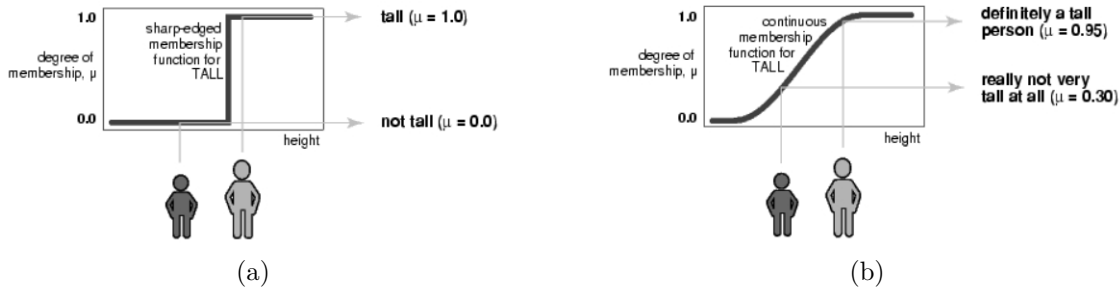


Figura 1.2: Diferencias entre la asociación de la altura a los conjuntos de alto y bajo mediante lógica clásica (a) y lógica difusa (b)

El uso de la lógica difusa permite establecer un grado de pertenencia a los conjuntos definidos, es decir las variables pueden pertenecer a más de un conjunto. Cuando estos términos se encuentran en estas representaciones entonces se tienen las **variables lingüísticas** donde estas en su totalidad conforman el **universo de discurso**, el proceso que se encarga de llevar las numéricas al dominio difuso (el de las variables lingüísticas) se llama **fuzzyficación**. Entonces al llevar los datos del dominio original ( $X$ ) al dominio difuso  $\tilde{A}$ , se tiene cada termino lingüístico corresponde a un **conjunto difuso**  $\tilde{a} \in \tilde{A}$  donde cada uno de estos conjuntos esta asociado a una función  $\mu$  tal que  $\mu_{\tilde{a}}(x) : X \rightarrow [0, 1]$ . Es decir que indican un grado de pertenencia a cada conjunto definido. Para el ejemplo anterior se pueden definir 2 conjuntos  $\tilde{A} = [\tilde{a}_1, \tilde{a}_2] = [pequeño, alto]$ , donde definiendo el mismo umbral de 1.70 mts se pueden obtener los siguientes resultados<sup>1</sup> para 1.30, 1.69, 1.90 [mts]:

- $\mu_A(1.50) = [0.99, 0.01]$ , donde esta muestra pertenece más al conjunto pequeño, que al alto.
- $\mu_A(1.69) = [0.495, 0.505]$ , donde esta tiene una pertenencia mixta a ambos conjuntos con valores similares.
- $\mu_A(1.90) = [0.01, 0.99]$ , donde esta tiene una pertenencia mayor al conjunto alto.

<sup>1</sup> Este corresponde a un ejercicio mental para ayudar la comprensión de la lógica difusa.

La lógica difusa opera en los llamados conjuntos difusos, donde los valores de están en un rango  $[0,1]$  en lugar del 0,1, donde estos valores son controlados por la **función de membresía** de un conjunto a cada elemento perteneciente al universo de discurso. Entonces el conjunto difuso  $A$  se caracteriza por su función de membresía  $\mu_A : X \rightarrow [0, 1]$  y se define como

$$A = \{(x, \mu_A(x))\} \quad (1.1)$$

### 1.2.2. Serie Temporal Difusa

Como su nombre lo indica, corresponde a series temporales que utiliza la lógica difusa para su procesamiento, convirtiendo cada valor de la serie en elementos pertenecientes a conjuntos difusos. Al tener una serie temporal en el dominio difuso, permite robustecer aun más el análisis sobre la serie, aplicando el análisis de relaciones de conjuntos difusos a los datos originales.

### 1.2.3. Operaciones difusas

Corresponden a las operaciones relacionadas a la lógica difusa, donde los elementos pertenecientes a una serie temporal difusa también pueden aplicarlo.

#### 1.2.3.1. Fuzzyficación.

Sea  $(A_i)_{i=1}^N$  una secuencia de conjuntos difusos y  $x \in X$  un valor de una serie temporal, entonces la **fuzzyficación** de  $x$  se define como.

$$fuzz(x) = (\mu_{A_i}(x))_{i=1}^N \quad (1.2)$$

#### 1.2.3.2. Intersección difusa.

Sean  $A$  y  $B$  2 conjuntos difusos definidos en  $X$  con funciones de membresía  $\mu_A$  y  $\mu_B$ , la intersección  $A \cap B$  se define por la función de membresía:

$$\mu_{A \cap B} = \min(\mu_A(x), \mu_B(x)) \quad (1.3)$$

#### 1.2.3.3. Reglas difusas e inferencia.

Los modelos de series difusas poseen un conjunto de reglas de inferencia difusas. La regla  $j$  tiene la forma dada por la ecuación[2]

$$\text{If } (X_n \text{ is } A_{j,0}) \wedge (X_{n-1} \text{ is } A_{j,1}) \wedge \dots \wedge (X_{n-m} \text{ is } A_{j,m}) \text{ then } (X_{n+1} \text{ is } A_{j,m+1}) \quad (1.4)$$

Donde  $X_t$  corresponde al valor de la serie de tiempo  $t \in [1, \dots, n, +1]$  y  $A_{j,k}$  son las variable lingüísticas a las cuales deben pertenecer para pertenecer para satisfacer los antecedentes de la regla.  $A_{j,m+1}$  es el consecuente o resultado de una inferencia. Teniendo esto, la regla establece que si los valores observados anteriores pertenecen a los conjuntos inferidos por la regla (en cuyo caso la regla se activa), se dice que el valor pronosticado de la serie de tiempo ( $X_{n+1}$ ) pertenecerá al conjunto  $A_{j,m+1}$ , con un valor de membresía igual a la magnitud en que los antecedentes son satisfechos. A este valor se le llama  $a_j$ .

Es posible encontrar situaciones donde más de una regla se active, para ello entonces se debe realizar el proceso contrario a la fuzzyficación de los datos, lo que se denomina **defuzzyficación**. Esto se debe aplicar a la data difusa para obtener un valor real del pronóstico. Para ello se puede utilizar el método del centro de gravedad. Este indica que si un conjunto de  $n$  reglas se activan con  $a_j$ , y el centro de la función de membresía triangular  $A_{j,m+1}$  es  $\beta_j$ , el valor pronosticado esta dado por la siguiente expresión:

$$X_{n+1} = \frac{\sum_j^n a_j \beta_j}{\sum_j^n a_j} \quad (1.5)$$

Cabe destacar que la generación de estas series dependen en gran medida de la configuración que se realice a las funciones que llevan a cabo la **fuzzyficación**, donde si esta configuración no es apropiada entonces la representación difusa de una serie temporal puede presentar mas desventajas que ventajas. Debido a estola creación de una serie temporal difusa debe ser un proceso muy estudiado y controlado, para ello el proceso de fuzzyficación sera llevado a cabo por funciones llamadas **particiones**, donde esta controla los grados de pertenencia de cada dato temporal, donde también permite la creación de **reglas** lingüísticas para mejorar el análisis de las series.

La creación de las series difusas corresponde a una parte del proceso de los modelos de series difusas, siendo estos los que se encargan de analizar los comportamientos temporales y difusos mediante las particiones y las reglas obtenidas.

## 1.3. Modelos de predicción

Son algoritmos computacionales cuya finalidad es obtener el comportamiento de una serie temporal en intervalos de tiempo que no poseen registros.

### 1.3.1. Modelos de machine learning.

Corresponden los sistemas más conocidos, no solo por su trabajo para realizar predicciones y regresiones de datos, sino también por su eficacia al clasificar datos con etiquetas asociadas (*machine learning supervisado*). Hay una gran cantidad de sistemas ya implementados que realizan este trabajo, pero se tendrá un enfoque en los siguientes modelos:

#### 1.3.1.1. Redes Neuronales Artificiales.

Es el algoritmo de Inteligencia Computacional más popular debido a su enorme aplicación en una gran cantidad de áreas. Corresponden a conjuntos de funciones matemáticas que ponderan los valores de una entrada para ajustarlo a un valor objetivo (salida), cada una de estas funciones se denominan *neuronas* y simplemente realizan una suma ponderada (figura 1.3a), donde el valor de estas sumas se procesa finalmente a una función para obtener su valor final. Esta última función se llama *función de activación* y permite a la neurona convertirse en un sistema no lineal.

Para que estos modelos puedan resolver problemas requiere el uso de varias neuronas, donde estas se agrupan por capas (figura 1.3b). El conjunto de todas estas capas dan lugar a una Red Neuronal Artificial, las cuales pueden ser de **entrada, oculta o salida** que indica si la capa esta al principio, en medio o final de la red respectivamente.

El algoritmo que realiza el proceso de aprendizaje de estos sistemas se llama *backpropagation*[3], el cual consiste en calcular el error obtenido en las ultimas capas y propagarlo hacia atrás, con el cual se realiza un ajuste de los valores de los pesos y bias (conexiones entre las neuronas y los valores independientes de esta) capa por capa hasta llegar al principio de la red. El entrenamiento de estos sistemas es un proceso iterativo, el cual se realiza durante una cantidad de **épocas** definidas, donde en cada una se realiza el cálculo de una función de costos o *loss*, que muestra el error del sistema alrededor del entrenamiento. El objetivo del algoritmo *backpropagation* es hacer disminuir lo mas que se pueda los valores que se obtengan de la función *loss*, es decir provocar una convergencia hacia el valor más pequeño posible (o en caso de que se tenga un métrica de eficacia del sistema, maximizar el valor de esta alrededor del entrenamiento).

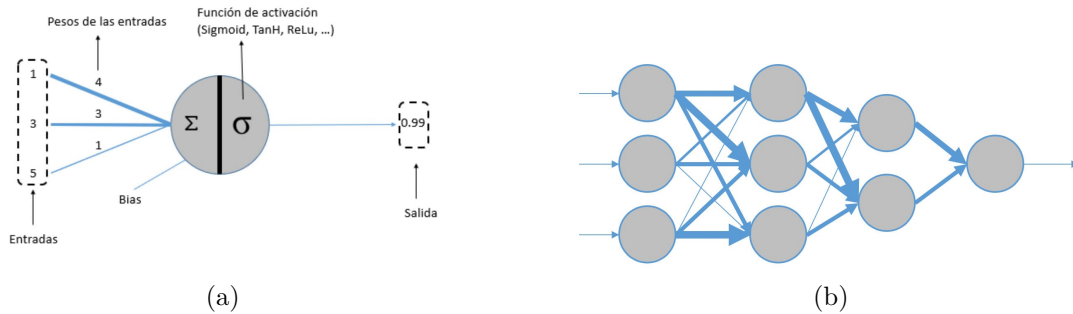


Figura 1.3: Diagrama de neurona artificial (a), y agrupación de estas en capas de una pequeña red (b)

El proceso de entrenamiento y la forma del modelo es controlado por los **hiperparámetros de la red**, los cuales configuran la forma de esta y también colocan pautas sobre como debe ser el entrenamiento que debe tener. Los más destacables son los siguientes:

- Tamaño de las capas ocultas: Cantidad de neuronas que utilizan las capas ocultas de la red.
- Función de activación: Corresponde al tipo de función no lineal que se aplica al resultado de la suma ponderada (los valores procesado por la capa anterior con los pesos de las conexiones de la capa actual). Se tienen diversas funciones, pero una de las más usadas corresponde a la función *RELU* la cual solo procesa valores positivos, es decir:

$$RELU(x) = \max(0, x) \quad (1.6)$$

- Cantidad de épocas: Corresponde al número de iteraciones que debe realizar el sistema durante el entrenamiento.
- Tasa de aprendizaje: Es un valor utilizado al momento de actualizar el valor que poseen los pesos, esta se encuentra incluida en la función que define el tipo de cambio que se le debe aplicar a estas conexiones (optimizador). Su valor es un parámetro muy importante ya que controla la convergencia del sistema, en caso de que sea un valor muy pequeño, el modelo requerirá una mayor cantidad de épocas de entrenamiento lo que se traduce en una ralentización del proceso de aprendizaje. Por otro lado si se escoge un valor muy alto, es posible que el modelo no pueda obtener los valores apropiados de los pesos, lo que conlleva a un mayor error al momento de calcular el *loss* de la época, por lo cual el modelo no lograr aprender correctamente.

- Optimizador: Corresponde al método por el cual se ajustan los valores de los pesos a partir del error obtenido. Un optimizador muy utilizado es el sistema **ADAM**, el cual utiliza estrategias de aprendizaje adaptativo para cambiar el valor de los pesos de la red junto con la tasa de aprendizaje, logrando de esta forma acelerar el aprendizaje junto con la convergencia de este proceso[4].

### 1.3.1.2. Random Forest.

Un modelo de *machine learning supervisado* muy conocido debido a sus buenos resultados en varias tareas de regresión y clasificación. Este sistema utiliza los conocidos arboles de decisión, siendo estos modelos que cambian el valor de un dato procesado dependiendo de las configuraciones que posea cada nodo del árbol, donde realiza *separaciones, hojas o splits* en cada nodo para extraer características e información importante de los datos procesados. El algoritmo Random Forest utiliza diversos arboles de decisión con diferente configuración cada uno (figura 1.4), donde los datos procesados pueden obtener diversos valores durante el procesamiento de cada árbol, sin embargo dependiendo del valor objetivo se descubre cual es la configuración óptima de todos estos arboles para procesar los datos de una forma más correcta, logrando así acercarse a los valores deseados.

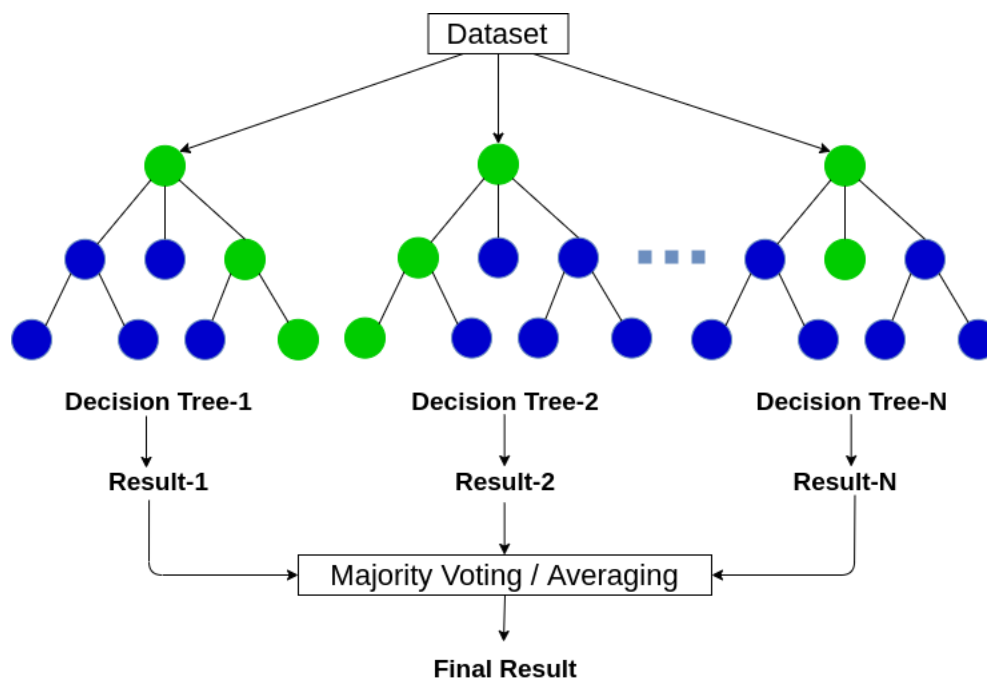


Figura 1.4: Diagrama del proceso de un algoritmo random forest.

Al igual que las Redes Neuronales, este sistemas también tiene sus hiperparámetros, que controlan la estructura de estos modelos. Los más destacados son:

- Cantidad de árboles: Corresponde al número de árboles de decisión que posee el algoritmo.
- Criterio de separación: Es la función que mide la calidad de los *splits* de los árboles. Un criterio destacado es la función Gini, la cual calcula el *índice Gini* el cual entrega la probabilidad de una característica específica que es clasificada incorrectamente cuando es seleccionada al azar. Esta definición indica que cuando este criterio es utilizado en los algoritmos Random Forest,

las características de la muestra que posean un menor valor serán más importantes. Este se puede obtener siguiendo la siguiente relación:

$$\text{Índice Gini} = 1 - \sum_{i=1}^n (p_i)^2 \quad (1.7)$$

Donde  $p_i$  corresponde a la probabilidad de asociación del elemento procesado a cada clase

- Cantidad de muestras mínimas para una separación: Es el número de muestras internas mínimo que son necesarios para realizar una separación de un nodo.
- Profundidad máxima: Largo máximo de los árboles, usualmente este parámetro se ajusta dependiendo del valor de la cantidad de muestras mínimas para una separación, ya que los nodos se expanden hasta que las hojas obtenidas contienen menos datos que este valor

### 1.3.2. Modelos de series temporales difusas.

Son los sistemas que utilizan lógica difusa para la predicción de datos en intervalos de tiempo que no tengan registros. Dado que la teoría de la lógica difusa es amplia y el análisis de las series temporales también lo es, entonces se tiene también la existencia de una gran cantidad de modelos difusos. Sin embargo a grandes rasgos, todos comparten una serie de características que permiten agruparlos[5].

#### 1.3.2.1. Parámetros relevantes

Son las características que se pueden configurar al utilizar los modelos difusos, los más destacados son:

- **Orden:** Este dato indica cuanta información pasada se necesita para obtener el valor siguiente, cabe destacar que a medida que se tome una mayor cantidad de datos previos, es más probable obtener un modelo con un mejor rendimiento (figura 1.5)

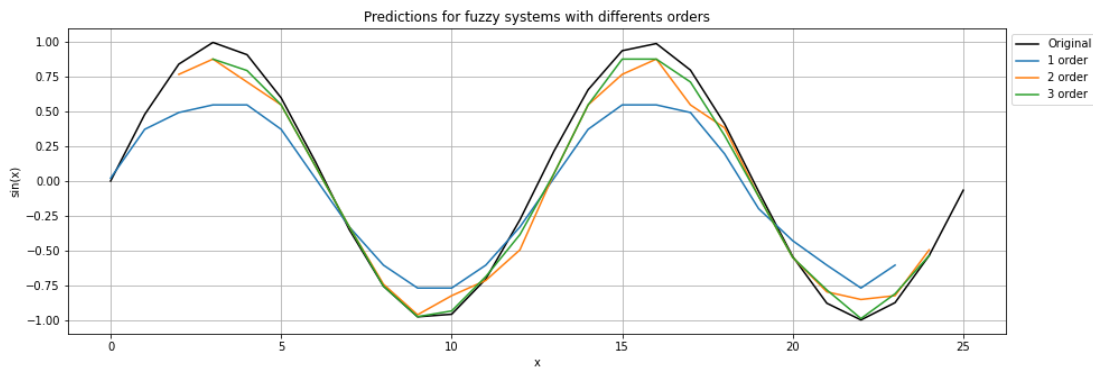


Figura 1.5: Comportamiento de predicciones de una función seno realizados por modelos difusos de diferentes ordenes.

- **Partición:** Es la función que controla la forma en la que los datos temporales son llevados a los conjuntos difusos, por ende se tienen 2 características relevantes en este parámetro: El

## tipo de partición y la cantidad de conjuntos difusos.

El primer término indica la forma general que tendrá la partición, algunos ejemplos son los siguientes:

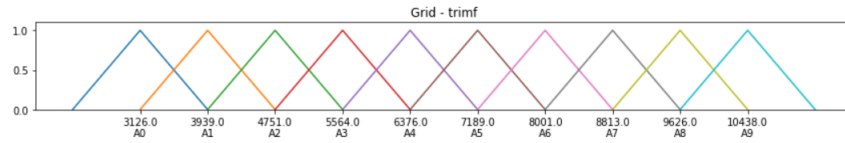


Figura 1.6: Partición Grid aplicado a 10 conjuntos difusos.

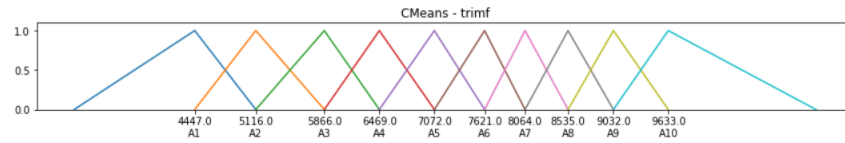


Figura 1.7: Partición tipo CMeans aplicado a 10 conjuntos difusos

Por otro lado la cantidad de conjuntos difusos como su nombre lo indica es el número de estos que estarán presentes en el proceso de fuzzyficación de la información. Cabe destacar la importancia de este parámetro, ya que al tener pocos conjuntos, entonces el sistema no logrará entrenarse de forma adecuada, en cambio si se tiene una gran cantidad conjuntos definidos, es posible que el sistema aprenda relaciones innecesarias dando como resultado un aumento del ruido y menor eficacia al momento de realizar las predicciones (similar a un sobreajuste de una red neuronal).

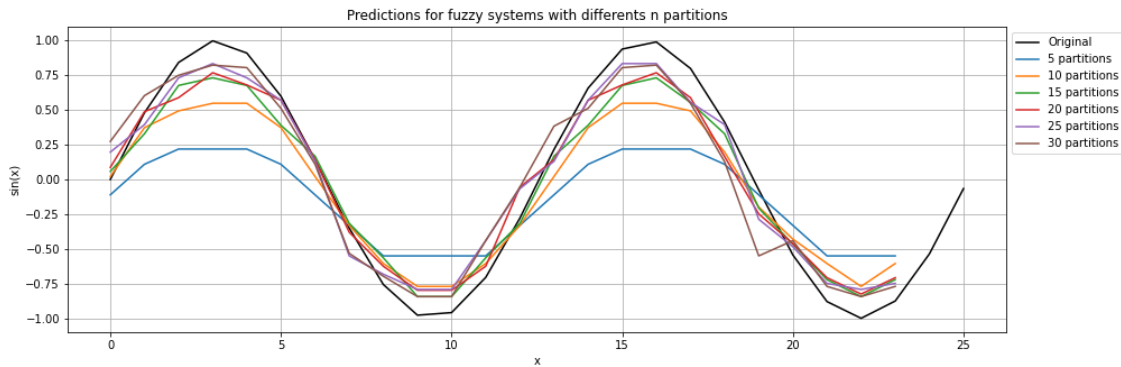


Figura 1.8: Predicciones de un modelo difuso para una función seno con distintas cantidades de particiones

- **Función de membresía:** Este parámetro controla el tipo de mapeo que se realice a la partición, por lo que controla la forma en la cual la data es asociada a conjuntos difusos



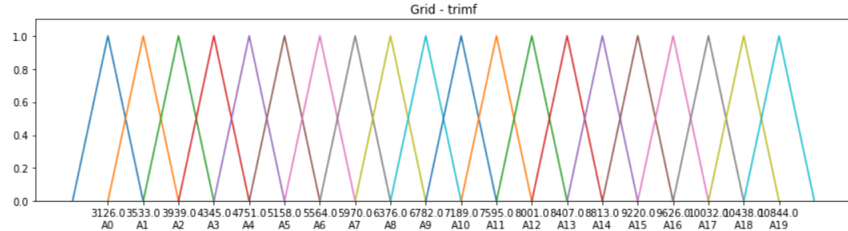


Figura 1.9: Partición Grid con función de membresía triangular aplicado a 20 conjuntos difusos.

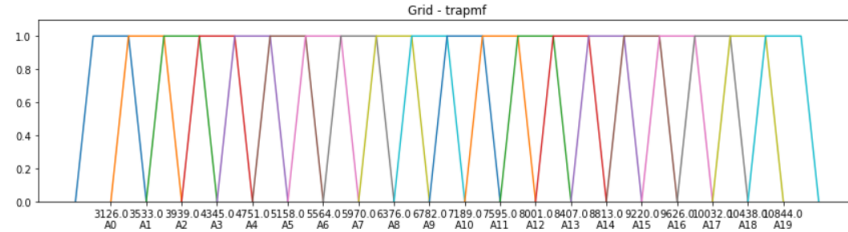


Figura 1.10: Partición Grid con función de membresía trapecio aplicado a 20 conjuntos difusos.

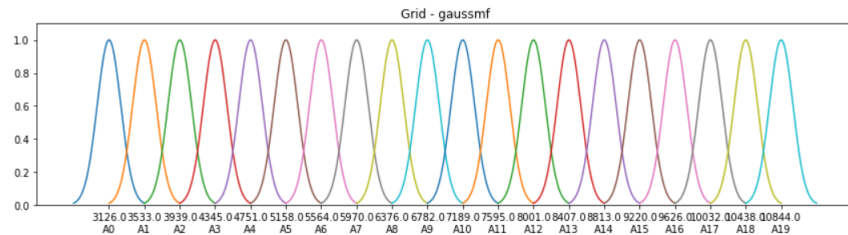


Figura 1.11: Partición Grid con función de membresía gaussiana aplicado a 20 conjuntos difusos.

Si bien se tiene una gran cantidad de particiones inspirados en diversas funciones matemáticas, la más utilizada y con mejores resultados corresponde a la partición Grid con función de membresía triangular (figura 1.9).

### 1.3.2.2. Modelos monovariantes y multivariantes.

Como la lógica difusa analiza los grados de pertenencia de las variables a conjuntos difusos, entonces la información a procesar puede tener datos no solamente de una variable objetivo, sino de también información relacionada a esta variable. Por ende se pueden categorizar estos modelos en monovariantes y multivariantes, donde los más destacados son los siguientes:

#### Modelos monovariantes

- **HOFTS** (High Order Fuzzy Time Series): Modelos cuyo principal parámetro para la creación de las reglas es el orden, no realiza ponderaciones de los conjuntos desarrollados, asumiendo que cada uno es igual de importante.

- **WHOFTS** (Weighted High Order Fuzzy Time Series): Modelo que tras la obtención de las reglas de partición, pondera los conjuntos difusos, por lo que se tendrán valores mas influyentes al momento de realizar las predicciones.
- **PWFTS** (Probabilistic Weighted Fuzzy Time Series): Modelo que tambien pondera la importancia de cada conjunto difuso, sin embargo este no solo tiene en consideración las reglas obtenidas, sino también la configuración inicial de la partición y el rendimiento de los modelos sin ponderaciones, por lo que la inclusión de estos pesos se realiza de manera probabilística. Cabe destacar que este modelo funciona mejor al tener una mayor cantidad de datos por lo que un aumento del orden puede ser contraproducente al utilizar este modelo.

## Modelos multivariantes

- **MVFTS** (MultiVariable Fuzzy Time Series): Es un sistema que analiza las relaciones entre las variables y no añade importancia a los conjuntos difusos creados, siendo similar al modelo HOFTS pero con la facultad de tener un análisis más robusto al obtener una predicción, debido a la presencia de información relacionada a la variable objetivo.
- **Weighted MVFTS**: Similar al modelo anterior, sin embargo le añade importancia a los conjuntos por lo que tiene un mejor análisis al desarrollar predicciones.
- **FIG-FTS** (Granular Multivariate Weighted Fuzzy Time Series): Son modelos que tienen un énfasis en la obtención de las relaciones de las variables mediante algoritmos de clusterización, además incorpora ponderaciones en los conjuntos difusos creados.

### 1.3.2.3. Proceso de un modelo difuso.

Para desarrollar predicciones mediante el uso de estos modelos se debe realizar el siguiente proceso[6]:

1. **Preprocesamiento de la data:** Es un paso genérico al utilizar modelos de predicción, aquí se realiza toda la limpieza y filtro de la información para llevarla a los modelos.
2. **Configuración de la partición:** Este es el **paso más importante** al utilizar los modelos difusos, se debe realizar un análisis preciso sobre la variable objetivo para saber cual seria el tipo de partición y cantidad de conjuntos difusos más apropiados. También se configura el tipo de modelo a utilizar y el orden de este.
3. **Fuzzyficación de la data:** En este paso, los datos son trasladados al dominio difuso, por lo que en este punto se generan las series temporales difusas.
4. **Generación de las reglas difusas:** En este punto se obtienen reglas de transición temporal, donde estas dependen de los parámetros con las que fue configurada la partición. Dependiendo del modelo, se obtiene la importancia de los conjuntos generados y las relaciones entre las variables procesadas
5. **Forecast:** Se realizan las predicciones con tomando una serie de valores cuyo largo debe ser de almenos el orden configurado en la partición, para poder realizar la predicción para los siguientes periodos.
6. **Defuzzyficación de la data:** Proceso contrario a la fuzzyficación, en este punto los datos vuelven a su dominio original.

## 1.4. Métricas de error

Al estar trabajando con regresores, se tienen las siguientes métricas par analizar la efectividad de las predicciones realizadas por los modelos.

### 1.4.1. RMSE

Es la simbología de **Root Mean Squared Error**, corresponde a la raíz del error cuadrático medio, dada una serie de valores  $[y_1, y_2, \dots, y_n]$  y una serie de predicciones  $[\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n]$  entonces la fórmula para el RMSE es:

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (1.8)$$

Donde este valor indica en términos del orden de los valores de la variable cual fue el error medio

### 1.4.2. MAPE.

Similar al anterior, el **Mean Absolute Percentage Error** mide el error medio porcentual del predicciones obtenidas. La obtención de esta se realiza a partir de la siguiente relación:

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{y_i - \hat{y}_i}{y_i} \quad (1.9)$$

Esta métrica cual fue el error porcentual analizando la relación entre el error y la data verdadera.

# Capítulo 2

## Base de datos.

### 2.1. Dataset.

La información utilizada en este trabajo son datos recopilados por la Red Europea de Gestores de Redes de Transporte de Electricidad (**ENTSO-E** por sus siglas en inglés), un organismo de gestores de electricidad que contiene registros sobre diversas variables relacionadas al mercado eléctrico de varios países europeos. Los datos mostrados corresponden a la información de consumo y generación española durante el 2015 y 2019<sup>2</sup>. El formato del dataset fue el siguiente:

time	Generation biomass [MW]	...	Forecast solar [MW]	...	Price Forecast [Eur/ MWh]	Price actual [Eur/ MWh]	Total load forecast [MW]	Total load actual [MW]
2015/01/01 00:00:00 + 01:00	447	...	17	...	50.1	65.41	26118	25385
2015/01/01 01:00:00 + 01:00	449	...	16	...	48.10	64.92	24934	24382
...	...	...	...	...	...	...	...	...
2018/12/31 23:00:00 + 01:00	290	...	26	...	64.27	69.88	24424	24455

Tabla 2.1: Información resumida disponible en la base de datos utilizada, debido a la gran cantidad de columnas, solo se mostraron estos datos.

En total se tiene un dataset con 29 características y 35064 muestras<sup>3</sup>, esta información contiene datos de la fecha registrada (en formato yyyy/mm/dd - hh/mm/ss), 21 fuentes de generación de energía (renovables y no renovables, los pronósticos de la generación de energías renovables en MW), el precio de la energía junto con pronósticos de esta (en EUR/MWh), y finalmente los datos de interés para este trabajo: **la demanda de la energía eléctrica horaria** junto con valores predichos por modelos del **TSO** español (Transmission System Operator, sociedad mercantil que

<sup>2</sup> Esta base de datos es pública y se puede obtener pulsando [aquí](#)

<sup>3</sup> Las columnas totales junto con una descripción breve de esta información se encuentra en el anexo: .

maneja la operación del sector eléctrico).

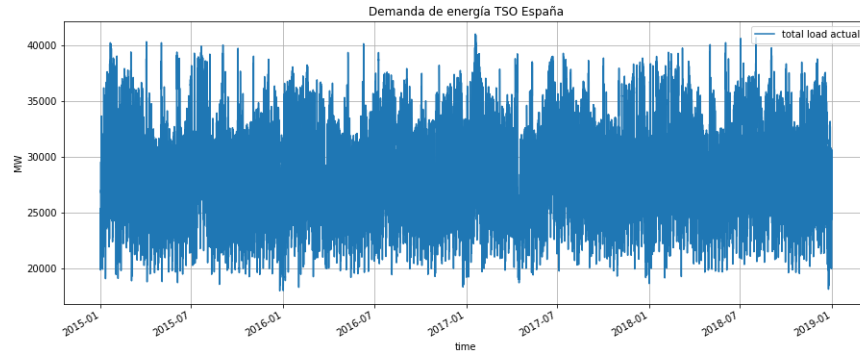
Los datos más utilizados fueron los de la demanda junto con los valores predichos por el TSO. Estos se utilizaron para comparar el rendimiento de los modelos con los sistemas utilizados por este órgano hasta 2019.

## 2.2. Análisis exploratorio de la demanda disponible.

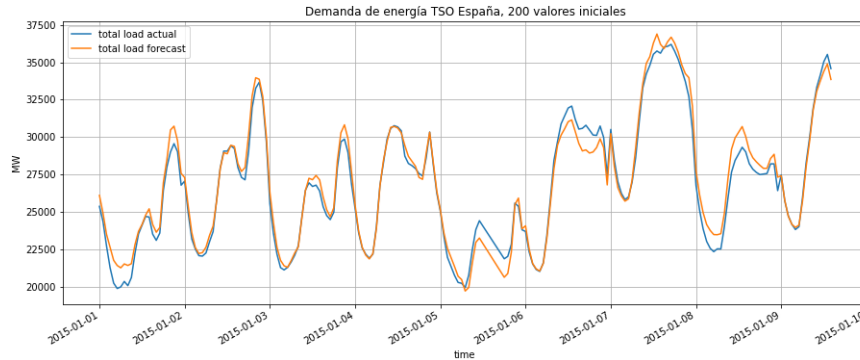
La información de mayor interés es la demanda, por lo que realizando un pequeño análisis se puede destacar lo siguiente:

Fechas registradas	Desde 01/01/2015 - 00:00:00 hasta 31/12/2018 - 23:59:59
Rango de la demanda	Mínimo: 18041 [MW], Máximo: 41015 [MW]
MAPE TSO forecast	1.096 %

Tabla 2.2: Datos de interés de la demanda disponible.



(a) Visualización de la demanda total



(b) Muestra de los primeros 200 valores de la demanda junto con el pronóstico

Figura 2.1: Visualización gráfica de la demanda total y los valores de esta durante los primeros 10 días de 2015.

# Capítulo 3

## Metodología.

Dada la información disponible en la base de datos (sintetizada en la tabla 2.1), se realizó un preprocesamiento de datos. Todo este proceso fue realizado en **python**, donde utilizando las funciones de la librería **pandas** se pudo obtener una visualización clara del dataset<sup>4</sup>.

EL primer filtrado realizado consistió en eliminar las columnas que tuvieron mas de un **90 %** de información nula<sup>5</sup> usando la función definida en el código 3.1, este pequeño algoritmo remueve las características de un **dataframe** cuyo porcentaje de valores nulos sea superior a un límite indicado (**nan\_threshold**). Entonces para realizar el proceso mencionado al principio, solo se cargó los datos en la entrada de esta función y se ajustó el valor de **nan\_threshold** a 0.9, donde además tambien se removieron las muestras que no tuvieron información en el resto de columnas restantes, esto con la finalidad de utilizar los datos relacionados a la demanda en los modelos difusos multivariantes. Este último filtro eliminó 47 muestras por lo que en este trabajo se utilizaron 35017 datos de generación, precio y demanda energética.

Tras la aplicación de esta limpieza de datos, se realizó un preprocesamiento para cada modelo que fuese probado, construyendo diversas series temporales o aplicando particiones a la data.

Código 3.1: Función de eliminación de columnas con poca información.

```
1 import pandas as pd
2
3 def nans_remove(df, nan_threshold, reset_index=True):
4     df_process = df.copy()
5     assert nan_threshold < 1 and nan_threshold > 0 , ('nan threshold must be between 0 and 1')
6     cols_names = df.columns
7     for name in cols_names:
8         len, drop_len = df_process[name].__len__(), df_process[name].dropna().__len__()
9         difference_reason = (len - drop_len)/(len)
10        if difference_reason > (1-nan_threshold):
11            df_process.drop(columns=[name], inplace=True)
12        else:
13            continue
14    df_process.dropna(inplace=True)
15    df_process.reset_index(drop=reset_index, inplace=reset_index)
16    return df_process
```

<sup>4</sup> La tabla 2.1 se obtuvo aplicando las funciones de esta librería

<sup>5</sup> Las columnas eliminadas fueron generation hydro pumped storage y forecast wind offshore day ahead

### 3.1. Preprocesamiento de datos.

Si bien se tiene una gran cantidad de información, los datos más relevantes son los relacionados a la demanda, por ende se seleccionaron los valores de esta junto con la predicción realizada por el TSO español, además de la incorporación del tiempo para ayudar el análisis de los resultados obtenidos.

time	total load actual	total load forecast
2015/01/01 00:00:00 + 01:00	25385	26118
2015/01/01 01:00:00 + 01:00	24382	24934
...	...	...
2018/12/31 22:00:00 + 01:00	25801	25450
2018/12/31 23:00:00 + 01:00	24455	24424

Tabla 3.1: Demanda registrada por el TSO español, junto con el pronóstico de esta y su fecha

Los datos de la demanda (**total load actual**) fueron procesados de diferente manera, dependiendo de cual fuese el modelo predictivo que se desee evaluar. Mientras que los valores del pronóstico solo se utilizaron para comparar la eficacia de los regresores con las predicciones realizadas por este órgano.

Para la Red Neuronal y Random Forest se tomaron los valores de la demanda y se construyeron diversas series temporales, como estos modelos necesitan una serie de características y una variable objetivo, entonces las características fueron los valores de la demanda previos a la actual.

time	load-24h	load-23h	...	load-1h	load actual
2015/01/02 00:00:00+01:00	25385	24382	...	26798	27070
2015/01/02 01:00:00+01:00	24382	22734	...	27070	24935
...	...	...	...	...	...
2018/12/31 22:00:00+01:00	29145	26934	...	28071	25801
2018/12/31 23:00:00+01:00	26934	24312	...	25801	24455

Tabla 3.2: Series temporales creadas a partir de los datos de la demanda junto con la fecha de registro.

Las series construidas tuvieron un largo de 25 valores, 24 valores previos de la demanda energética junto con el valor actual, por lo que los valores de la demanda de todo el primer día fueron perdidos (no se tuvo información previa). Con esta información se definió lo siguiente:

- Características del dataset: Demanda registrada durante las últimas 24 horas. Se escogió esta cantidad de características debido a que la información de la demanda fue registrada horaria.
- Variable objetivo: Valor actual de la demanda.

Por otro lado, los modelos de series difusas trabajaron directamente con la serie temporal de la demanda, convirtiéndola en conjuntos difusos para diferentes configuraciones de particiones.

## 3.2. Configuración de modelos.

### 3.2.1. Modelos de machine learning.

Como se mencionó anteriormente, se utilizaron 2 modelos: Una Red Neuronal Artificial y un Random Forest. Los hiperparámetros de la red utilizada fueron:

Hiperparámetro	Valor
Cantidad de capas ocultas	1
Tamaño de la capa oculta	100 neuronas
Función de activación de la capa oculta	RELU
Cantidad de épocas	200
Tasas de aprendizaje	0.001
Optimizador	ADAM

Tabla 3.3: Hiperparámetros de la Red Neuronal utilizada.

Estos valores muestran la construcción de una red sencilla. Por otro lado los hiperparámetros del sistema Random Forest fue el siguiente:

Hiperparámetro	Valor
Cantidad de árboles	100
Criterio de separación	Índice de Gini
Cantidad de muestras mínimas para una separación	
Profundidad máxima	Ajustable

Tabla 3.4: Hiperparámetros del sistema Random Forest utilizado.

El sistema Random Forest utilizó muchos valores que vienen por default en la librería **sklearn**. Hay que destacar que las profundidades máximas de los árboles fueron un valor ajustado, ya que estos se extendieron hasta que las hojas finales tuvieron menos valor que el configurado en la cantidad de muestras mínimas para una separación.

### 3.2.2. Modelos de series difusas.

Para el desarrollo de los procesos mencionados se utilizó la librería **pyFTS**, la cual desarrolla de manera precisa todo lo relacionado a series temporales difusas.

#### 3.2.2.1. Partición.

Se utilizó una partición general: **Grid Partition**, la cual distribuye equitativamente el grado de pertenencia según la cantidad de conjuntos difusos creados (figura 1.9). Para definir la cantidad de conjuntos difusos se utilizó lo siguiente:

- Al estar trabajando con valores de demanda de energía, entonces se crearon **5 niveles**: Muy Bajo, Bajo, Medio, Alto y Muy Alto, o por sus siglas en inglés **VL, L, M, H, VH**



- Cada nivel se decidió dividir a su vez en 7 subniveles, por ende los conjuntos difuso generados tuvieron nombres [VL0, VL1, ... VH5, VH6], lo que dió un total de 35 conjuntos.

Para las funciones de membresía, se utilizaron 3 tipos: **triangular**, **trapecio** y **gaussiana**, siendo la función triangular la más utilizada.

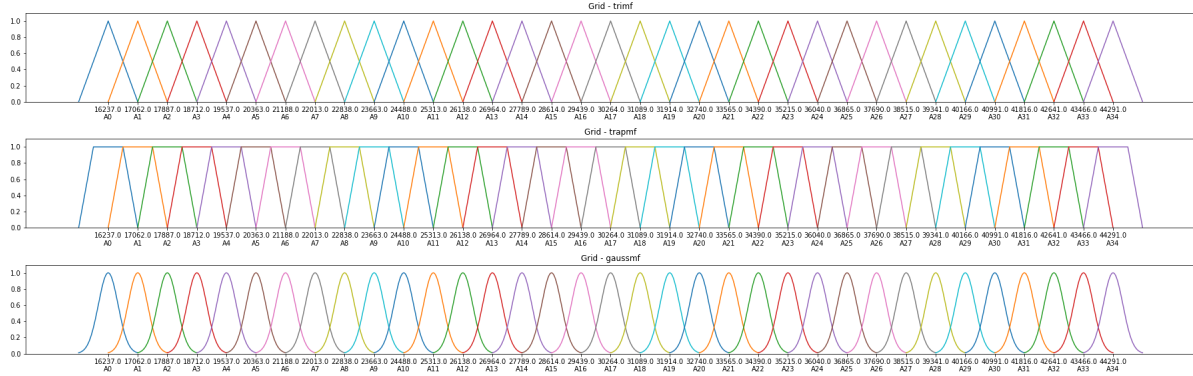


Figura 3.1: Visualización de la partición Grid configurada con 35 niveles y 3 tipos de funciones de membresía. El eje y representa el grado de pertenencia de algún valor temporal (eje x).

## Partición temporal

Hay que destacar que las particiones mostradas en la figura 3.1 se aplicaron a los datos de la demanda en los modelos monovariados. Al utilizar los modelos multivariados se tuvo que aplicar una partición diferente, los datos relacionados fueron las fechas, por lo que se aplicó una partición temporal a estos datos.

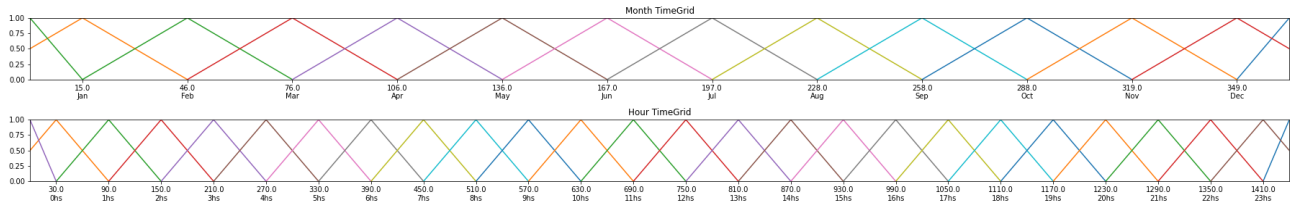


Figura 3.2: Partición temporal aplicado a las fechas registradas.

Se aplicaron 2 tipos de particiones: una mensual con 12 conjuntos difusos y una horaria con 24 conjuntos generados. Ambas particiones utilizaron una función de membresía triangular.

### 3.2.2.2. Orden.

Se probaron los modelos hasta el 3er orden, donde la extracción de las variables previas se realizó de manera automática debido a los objetos configurados en la librería **pyFTS**.

### 3.2.2.3. Tipos de modelos.

Como se ha mencionado, se programaron modelos monovariados y multivariados, donde cada uno de estos tuvo una partición con una función de membresía específica, orden con presencia de métodos de ponderación de los conjuntos para alguno de estos modelos.

Modelo	Partición	Función de membresía	Orden
HOFTS	Grid Partition, sin ponderación	triangular, trapecio, gaussiana	1, 2 y 3
WHOFTS	Grid Partition, con ponderación	triangular, trapecio, gaussiana	1, 2 y 3
PWFTS	Grid Partition, con ponderación probabilística	triangular, trapecio, gaussiana	1, 2 y 3

Tabla 3.5: Características generales de los modelos monovariantes usados, las ponderaciones mencionadas hacen referencia al cálculo de importancia de los conjuntos difusos generados.

Los modelos multivariantes siguen una estructura similar a los modelos anteriores, sin embargo estos incorporan cálculos de las relaciones entre las variables a procesar, donde se debe destacar la variable objetivo en donde para todos los modelos fue la demanda.

Modelo	Características generales
MVFTS	Grid Partition triangular de orden 3 sin ponderación de conjuntos difusos, con particiones mensuales y horarias a las fechas registradas.
Weighted MVFTS	Grid Partition triangular de orden 3 con ponderación de conjuntos difusos, con particiones mensuales y horarias a las fechas registradas.
FIG-FTS	Grid Partition triangular de orden 2, con knn como algoritmo de clusterización para búsqueda de las relaciones, con $k = [1,2,3]$

Tabla 3.6: Configuraciones generales de los modelos multivariantes, la partición inicial fue la aplicada a la variable objetivo.

### 3.3. Entrenamiento

Se tomó un 70% del total de muestras para el conjunto de entrenamiento, por ende la distribución de datos fue la siguiente:

Tipo de modelo	Muestras de entrenamiento	Muestras de prueba
ML supervisado	24495	10498
Difusos	24511	10506

Tabla 3.7: Cantidad de muestras de conjuntos por modelo

Se tiene que los modelos difusos utilizaron una pequeña mayor cantidad de datos, debido a que no se realizó el proceso de construcción de series, ya que estos modelos no obedecen a la creación de características y objetivo.

Se utilizaron las librerías **sklearn** y **pyFTS** para la configuración y entrenamiento de los modelos supervisados y difusos respectivamente, donde una vez los modelos estuvieron ajustados para funcionar con las especificaciones mencionadas en la sección anterior, se entrenaron. Durante este proceso se aplicó el algoritmo de *backpropagation* en las redes neuronales, búsqueda óptima de las separaciones en el sistema Random Forest, y la búsqueda de las reglas de inferencia óptimas de predicción para los modelos difusos.

# Capítulo 4

## Resultados.

La principal métrica utilizada en esta sección para evaluar el rendimiento de los modelos es el **MAPE**, en donde se aplicó tras obtener las predicciones de cada modelo entrenado con la información de los conjuntos de prueba.

### 4.1. Modelos de ML.

Aplicando estos modelos para predecir el comportamiento de la demanda entre el 27 y 28 de Octubre del 2017 se obtuvieron los siguientes comportamientos.

#### 4.1.1. Red Neuronal

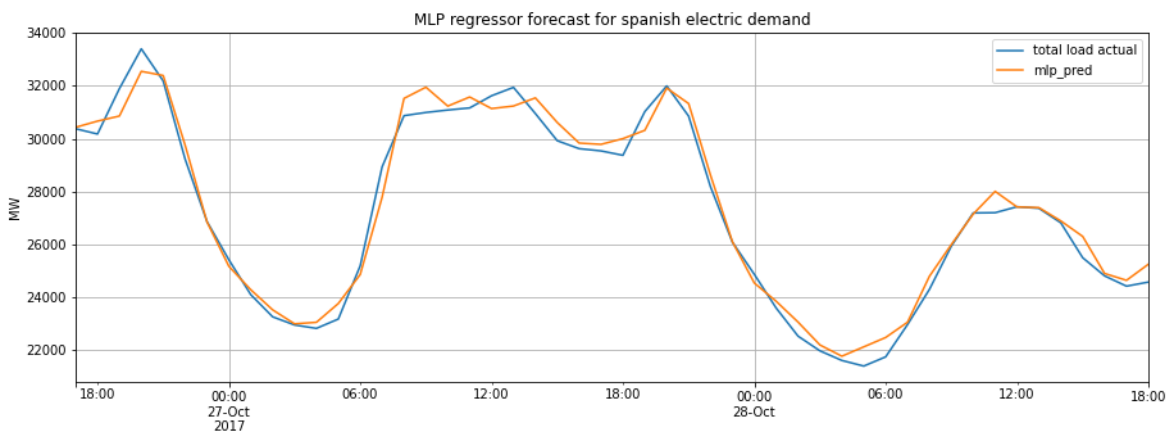


Figura 4.1: Predicciones de la demanda eléctrica de la red neuronal descrita en la sección 3.2.1

Durante estas fechas el sistema tuvo una varios aciertos de la demanda, y este comportamiento se mantuvo similar hasta el final de las fechas registradas. El MAPE para este modelo fue de 1.5371 %

### 4.1.2. Random Forest

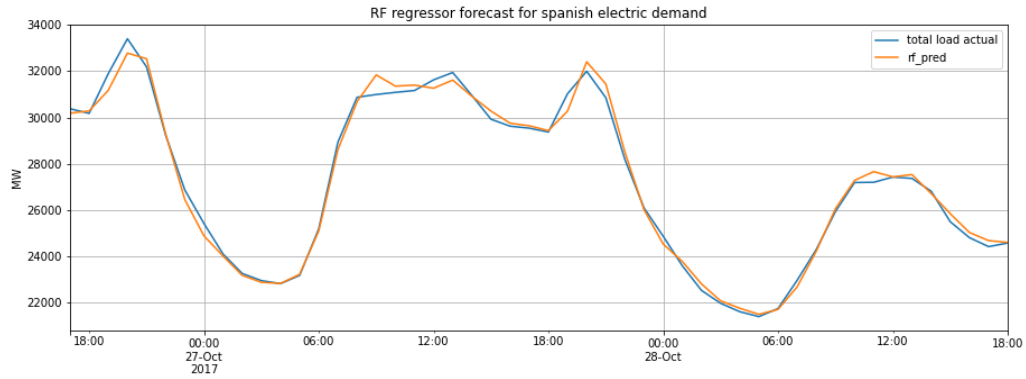


Figura 4.2: Predicciones de la demanda eléctrica de la Random Forest descrito en la sección 3.2.1

Este sistema tuvo mas aciertos que el anterior y su comportamiento fue parecido hasta el final de la serie. El MAPE obtenido con este modelo fue de 1.07 %.

### 4.1.3. Rendimiento de modelos de ML

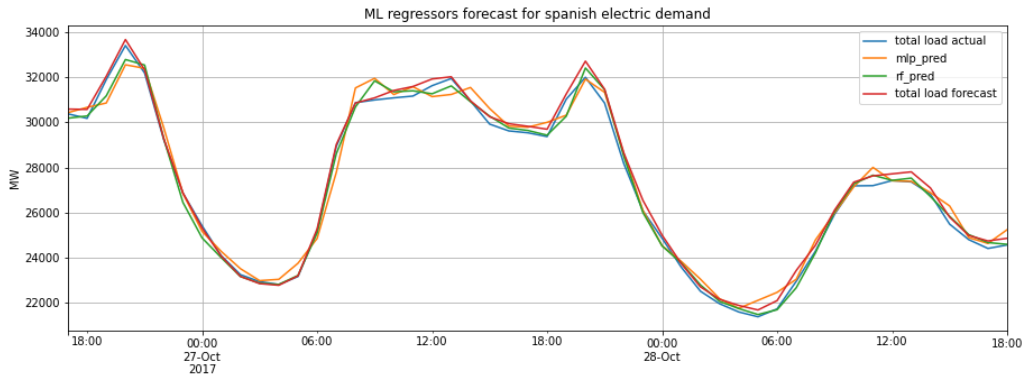


Figura 4.3: Comparación de las predicciones de los modelos junto con la predicción realizada por el TSO español

Sintetizando los rendimientos obtenidos por estos modelos se tienen los siguientes valores.

Modelo	MAPE
TSO forecast	0.9590
Random Forest	1.5371
Random Forest	1.0700

Tabla 4.1: Rendimiento de modelos de machine learning supervisados junto con el MAPE obtenido por el TSO

## 4.2. Modelos de series difusas.

Los modelos difusos definidos en el capítulo anterior fueron aplicados al conjunto de prueba, donde las visualizaciones de la demanda fue entre el 22 y 23 de Diciembre del 2019. Cabe destacar que este fueron los resultados de la aplicación del Grid Partition triangular (figura 3.1)

### 4.2.1. Modelos monovariables.

#### 4.2.1.1. HOFTS

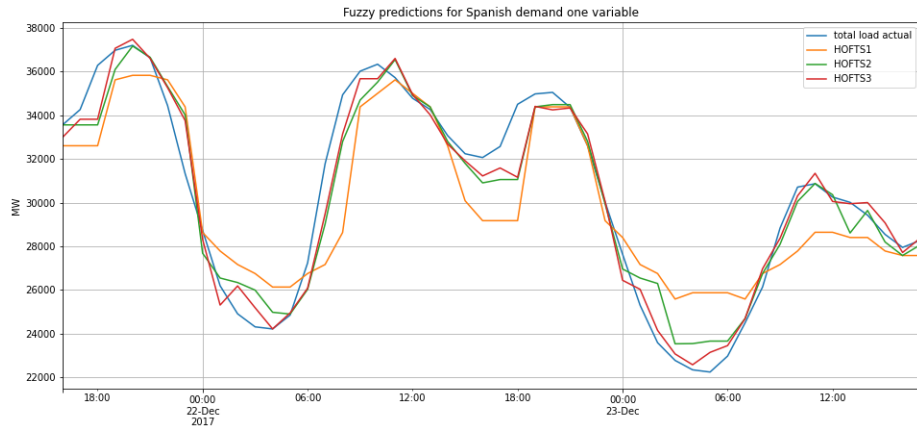


Figura 4.4: Predicciones de la demanda realizado por el modelo monovariable HOFTS para 3 ordenes

Este modelo obtuvo un MAPE igual a 6.90 %, 3.61 % y 2.94 % para los ordenes 1, 2 y 3 respectivamente

#### 4.2.1.2. WHOFTS

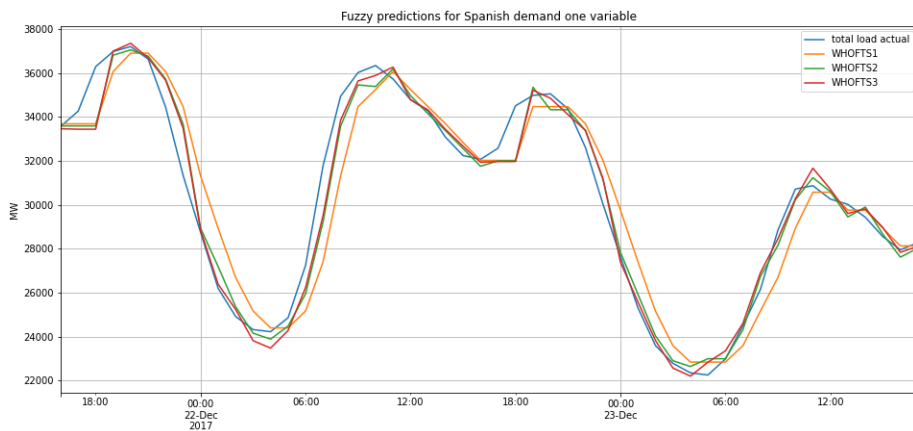


Figura 4.5: Predicciones de la demanda realizado por el modelo monovariable WHOFTS para 3 ordenes

Este modelo tuvo más aciertos que el HOFTS, los MAPES obtenidos fueron de 3.89 %, 2.54 % y 2.33 % para los ordenes 1, 2 y 3 respectivamente.

#### 4.2.1.3. PWFTS

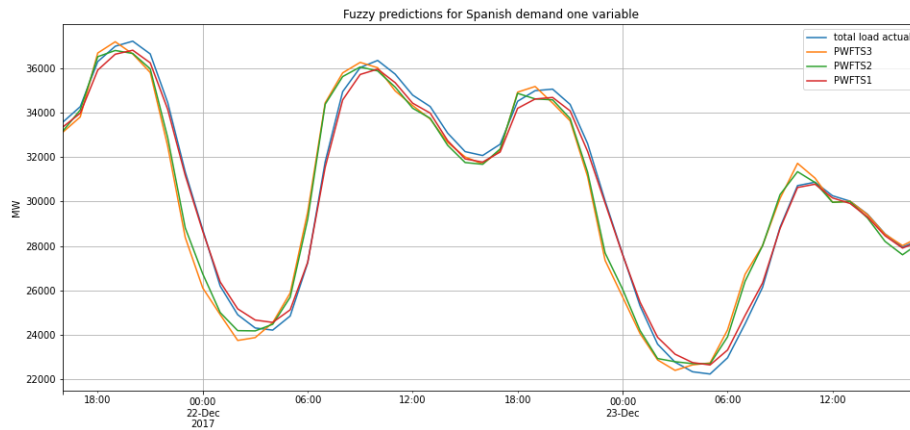


Figura 4.6: Predicciones de la demanda realizado por el modelo monovariable PWFTS para 3 ordenes

Los aciertos aumentaron notoriamente con el uso de este sistema, a tal punto que el modelo de orden 1 tiene una gran similitud con la demanda real. Los MAPE obtenidos fueron de 0.87 %, 2.46 % y 2.83 % para los órdenes 1, 2 y 3 respectivamente.

#### 4.2.1.4. Rendimiento de modelos de una variable.

Ordenando los modelos según el MAPE obtenido se obtiene la siguiente tabla:

Modelo	Orden	MAPE ( %)
PWFTS	1	0.8783
TSO forecast	No aplica	0.9590
WHOFTS	3	2.3359
PWFTS	2	2.4677
WHOFTS	2	2.5408
PWFTS	3	2.8389
HOFTS	3	2.9445
HOFTS	2	3.6125
WHOFTS	1	3.8930
HOFTS	1	6.9030

Tabla 4.2: Error Porcentual Medio Absoluto obtenido por los modelos difusos monovariables.

Se tiene que el modelo PWFTS supera a las predicciones realizadas por el TSO, además de un mejor rendimiento general para los modelos difusos de mayor orden (exceptuando el PWFTS).

## 4.2.2. Modelos multivariantes.

Incorporando la partición temporal a las fechas e incluyéndolas en el proceso de predicción se obtuvieron los siguientes comportamientos.

### 4.2.2.1. MVFTS

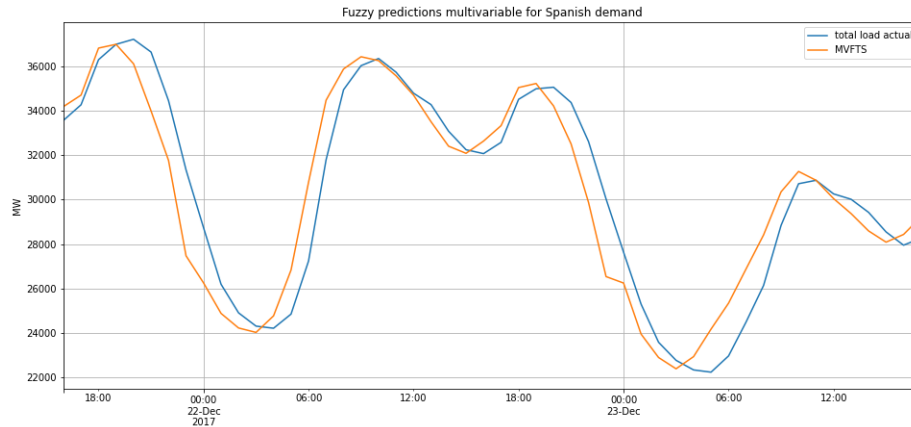


Figura 4.7: Predicciones de la demanda realizado por el modelo multivariable MVFTS

Este sistema tuvo un MAPE de 3.60 %

### 4.2.2.2. Weighted MVFTS

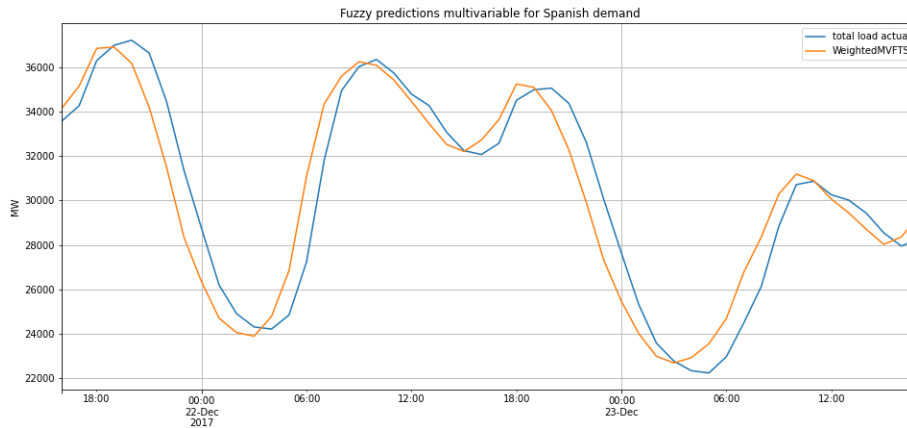


Figura 4.8: Predicciones de la demanda realizado por el modelo multivariable WeightedMVFTS

Incorporando las ponderaciones de en los conjuntos difusos se mejora parcialmente el rendimiento, este sistema obtuvo un MAPE de 3.48 %



#### 4.2.2.3. FIG - FTS

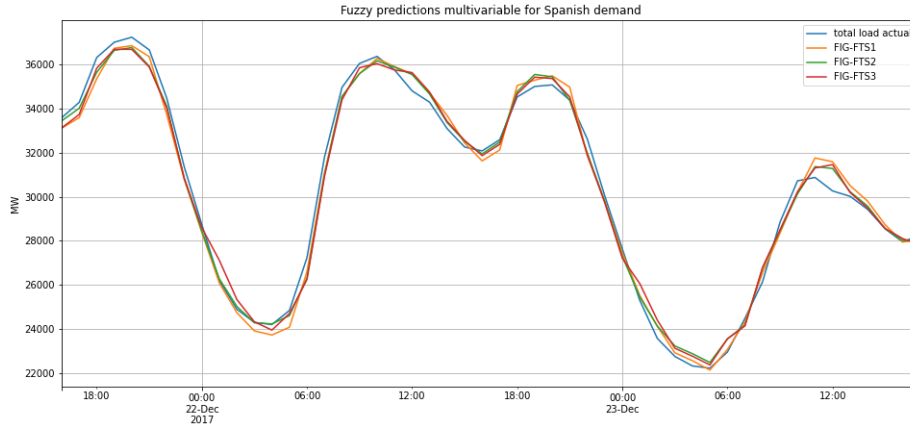


Figura 4.9: Predicciones de la demanda realizado por el modelo multivariable FIG-FTS, hasta 3 vecinos en el uso de KNN.

La búsqueda de agrupaciones utilizando algoritmos de clusterización de machine learning mejora el rendimiento, se obtuvo un MAPE de 1.70 %, 1.54 % y 1.58 % para 1, 2 y 3 vecinos en el uso de knn para el cálculo de relaciones entre las variables.

#### 4.2.2.4. Rendimiento de modelos multivariables.

Ordenando los modelos segun el MAPE obtenido como en la sección anterior se obtiene lo siguiente:

Modelo	MAPE (%)
TSO forecast	0.9590
FIG - FTS (k=2)	1.5421
FIG - FTS (k=3)	1.5866
FIG - FTS (k=1)	1.7079
WeightedMVFTS	3.4820
MVFTS	3.6086

Tabla 4.3: Error Porcentual Medio Absoluto obtenido por los modelos difusos multivariables.

En este caso no se tiene un modelo que tuvo un rendimiento superior al TSO, sin embargo las predicciones de estos sistemas en general tuvieron un error inferior al de los modelos monovariables.

#### 4.2.3. Rendimiento General.

Los resultados mostrados anteriormente solo usaron una función de membresía triangular, se realizaron pruebas también con otro tipo de función, ordenando el rendimiento obtenido por esos modelos junto con los resultados de las secciones anteriores se obtiene el siguiente ranking de modelos.

Modelo	Tipo	Orden	Partición	MAPE (%)
PWFTS	single	1	Grid Partition tri	0.8738
PWFTS	single	1	Grid Partition gauss	0.8779
PWFTS	single	1	Grid Partition trap	0.8851
TSO forecast	No aplica	No aplica	No aplica	0.9590
FIG-FTS (k=2)	multi	2	Grid Partition tri	1.5421
FIG-FTS (k=3)	multi	2	Grid Partition tri	1.5866
FIG-FTS (k=1)	multi	2	Grid Partition tri	1.7079
WHOFTS	single	3	Grid Partition tri	2.3359
WHOFTS	single	3	Grid Partition trap	2.3359
PWFTS	single	2	Grid Partition trap	2.3936
PWFTS	single	2	Grid Partition tri	2.4677
WHOFTS	single	2	Grid Partition tri	2.5408
WHOFTS	single	2	Grid Partition trap	2.5408
WHOFTS	single	3	Grid Partition gauss	2.6005
PWFTS	single	2	Grid Partition gauss	2.6868
PWFTS	single	3	Grid Partition trap	2.7710
PWFTS	single	3	Grid Partition tri	2.8389
WHOFTS	single	2	Grid Partition gauss	2.9332
HOFTS	single	3	Grid Partition trap	2.9445
HOFTS	single	3	Grid Partition tri	2.9445
PWFTS	single	3	Grid Partition gauss	3.0015
WeightedMVFTS	multi	2	Grid Partition tri	3.4820
HOFTS	single	3	Grid Partition gauss	3.6079
MVFTS	multi	1	Grid Partition tri	3.6086
HOFTS	single	2	Grid Partition trap	3.6125
HOFTS	single	2	Grid Partition tri	3.6125
WHOFTS	single	1	Grid Partition tri	3.8930
WHOFTS	single	1	Grid Partition trap	3.8930
WHOFTS	single	1	Grid Partition gauss	3.9295
HOFTS	single	2	Grid Partition gauss	4.7830
HOFTS	single	1	Grid Partition tri	6.9030
HOFTS	single	1	Grid Partition trap	6.9030
HOFTS	single	1	Grid Partition gauss	8.0617

Tabla 4.4: Modelos difusos probados junto con su rendimiento

En la gran mayoría de modelos, cuando solo se cambió la función de membresía el rendimiento no cambió mucho, por ende se espera tener un comportamiento gráfico similar al mostrado en las secciones anteriores.

### 4.3. Análisis

Los modelos de machine learning supervisado tuvieron rendimientos elevados al realizar las predicciones, esto fue posible debido a la construcción de las series temporales de la demanda tomando un periodo de 24 muestras. Tanto la Red Neuronal como el Random Forest pudieron encontrar más características al analizar estas series debido a que la demanda tiene un comportamiento relativamente periódico. Sin embargo el único modelo que logro superar a los pronósticos de la demanda del TSO fue el PWFTS, debido a la forma de extraer las reglas ponderando probabilísticamente los conjuntos difusos.

Analizando los rendimientos obtenidos en la tabla 4.4 se aprecia que las funciones de membresía no afecta mucho el rendimiento de los modelos difusos, en parte debido a que el poder de predicción se basa en la generación de las reglas de inferencia por lo que el mapeo de la partición configurada no es un parámetro muy relevante, aunque en algunos casos si se tuvieron cambios al utilizar diferentes funciones. Sin embargo este impacto pierde fuerza al analizar los modelos con diferente orden, dando a entender el peso de este parámetro al momento de realizar los entrenamientos (todos los modelos a mayor orden tuvieron un mejor rendimiento exceptuando el PWFTS).

Los modelos difusos que no ponderaron los conjuntos fueron los que tuvieron un rendimiento menor, esto muestra la gran importancia que tiene esta característica al momento de configurar los modelos ya que se tendrán conjuntos difusos mas relevantes al momento de generar las reglas de inferencia y posteriormente realizar las predicciones de la demanda.

# Capítulo 5

## Conclusiones.

Se logró encontrar un modelo que pudo superar el rendimiento predictivo de un órgano destacado como el TSO, sin embargo aun se debe analizar bien el resultado de este comportamiento ya que el modelo PWFTS más eficaz fue el de menor orden, siendo este último un parámetro sumamente relevante al realizarse el entrenamiento de los sistemas difusos.

La teoría indicaba la poca relevancia que tenía la función de membresía, y la importancia del orden al momento de configurar un modelo de serie temporal difusa, por lo que los resultados obtenidos apoyan esto. Sin embargo se obtuvieron casos en donde si hubo diferencia notable entre mismos modelos pero con distinta función de membresía, por lo que se requiere un análisis minucioso de estos eventos para llegar a las causas de estos comportamientos.

Hasta el momento, una de las mayores dificultades ha sido el poder utilizar modelos que posean un orden elevado, los resultados logrados indican un mejor rendimiento al subir el orden, sin embargo al aumentar el valor de esta característica también se hace más costoso computacionalmente. Esta misma consecuencia sucede al incorporar variables relacionadas a la demanda en los modelos multivariados, por ende se deben buscar formas de implementar modelos con mejores valores en sus parámetros sin aumentar de forma drástica el costo computacional.

Las series difusas han mostrado su eficacia en este trabajo al desarrollar modelos predictivos de la demanda de energía eléctrica, sin embargo el modelo final aun no ha sido construido hasta la fecha, ya que se requiere se desee incorporar las variables estocásticas relacionadas a la demanda para fortalecer el modelo, donde lo más lógico es incorporarlo a los modelos multivariados, sin embargo el hecho de obtener resultados que ya superan predicciones muy exactas indican que se lleva un buen rumbo en el trabajo desarrollado, pero es pronto decir que se ha logrado obtener un modelo que puede competir con los del estado del arte, ya que para ello se deben replicar los modelos de la literatura con el modelo final obtenido y realizar una comparación con los resultados de las predicciones. Dado esto como trabajo futuro se deben realizar una réplica de dichos modelos para analizar su rendimiento con los valores de la demanda energética utilizados en este trabajo.

# Bibliografía

- [1] E. Arriols, “Cuales son las fuentes de energía más utilizadas en el mundo,” 2020.
- [2] C. L. J. J. Flores, “Aprendizaje de modelos difusos para predicción de series de tiempo,” Octubre 2015.
- [3] S. Kostadinov, “Understanding backpropagation algorithm,” Agosto 2019.
- [4] M. W. S. Bock, *A Proof of Local Convergence for the Adam Optimizer*. Rengesburgh, Alemania: International Joint Conference, Julio 2019.
- [5] P. Silva, “A short tutorial on fuzzy time series,” Diciembre 2018.
- [6] M. Intelligence and D. S. Laboratory, *pyFTS Quick Start*. 2018.