

Universidad de Cienfuegos “Carlos Rafael Rodríguez”

Facultad de Ingeniería

Carrera de Ingeniería Informática

19-6-2019



UNIVERSIDAD  
DE CIENFUEGOS

# MODELO PARA PREDECIR EL CONSUMO DE ENERGÍA ELÉCTRICA DE UNA EDIFICACIÓN HOTELERA BASADO EN TÉCNICAS DE MINERÍA DE DATOS

**Autor:**

Jorge David García Tamayo.

**Tutores:**

MSc. Yaillem Arencibia Rodríguez del Rey

Dr. Boris Gabriel Vega Lara

Lic. Julio Camejo Corona

**Cienfuegos, Cuba**

## Agradecimientos

Primeramente, a Dios por darme la posibilidad de vivir este momento único y por brindarme el entendimiento y la sabiduría para afrontar los obstáculos que fueron apareciendo durante el trayecto.

A mi esposa Sarahy, por estar siempre para mí, por aconsejarme sabiamente en cada encrucijada, por poner siempre su amor y paciencia delante de las dificultades, y por creer en mí.

A mi hija querida, por haberme convertido en otro yo, por darme la felicidad de ser padre y por brindarme la posibilidad de contemplar su bella sonrisa, no importando las circunstancias.

A mis tutores Yailem y Julio por ser siempre tan especiales, tan atentos y detallistas; y que, aunque a veces uno discordaba del otro, siempre buscaban la forma de acomodarlo todo con el fin de ayudarme a vencer esta linda pero tortuosa etapa de mi vida.

A mis padres por siempre estar presentes, pese a la distancia; porque en momentos como estos siempre se mantienen atentos a mí, y porque lo que soy se lo debo a ellos.

A mis suegros por apoyarme siempre.

A mis compañeros del grupo por estar ahí en las buenas y las malas durante estos preciosos cinco años, por brindarme cada día esos momentos de alegría, pero también de reflexión que disfruté tanto.

A todos mis demás familiares, amigos y profesores, porque cada uno desde su posición brindó lo que pudo, y muchas veces fue más de lo que necesité. Gracias a todos.

## Resumen

La presente investigación titulada “**MODELO PARA PREDECIR EL CONSUMO DE ENERGÍA ELÉCTRICA DE UNA EDIFICACIÓN HOTELERA BASADO EN TÉCNICAS DE MINERÍA DE DATOS**”, constituye una alternativa de eficiencia energética para predecir el consumo de energía de una edificación a partir de variables climatológicas y de ocupación del hotel. La obtención de un modelo de predicción, basado en técnicas de minería de datos, brinda a los especialistas en energía una herramienta de apoyo en la toma de decisiones.

La metodología de minería de datos KDD fue la seleccionada para desarrollar la experimentación y las técnicas de minería de datos más apropiadas resultaron ser: ANN, SVM y los DT.

La experimentación se desarrolla en Weka y MATLAB, consiste en probar con diferentes variantes para cada técnica, modificando parámetros de la arquitectura y el entrenamiento.

Se experimentaron con 225 variantes del MLP, 90 variantes del FFNN, 45 variantes del LRN, 45 variantes del GRNN, 45 variantes del NARX, 45 variantes del DT y 45 variantes del SVM.

Se proponen dos modelos: para temporada alta el DT (creado, entrenado y probado con una semilla de generación aleatoria igual a 3 y el dataset de temporada alta sin selección de atributos) con una efectividad de 93.99 %; y para temporada baja el DT (creado, entrenado y probado con una semilla de generación aleatoria igual a 1 y el dataset de temporada baja sin selección de atributos) con una efectividad de 96.61 %. Para interactuar con estos modelos se desarrolló una interfaz gráfica en MATLAB.

**Palabras clave:** modelo, predicción, energía eléctrica, edificación.

## Summary

The present research entitled "MODEL TO PREDICT THE ELECTRICITY ENERGY CONSUMPTION OF A HOTEL BUILDING BASED ON DATA MINING TECHNIQUES", constitutes an alternative of energy efficiency to predict the energy consumption of a building based on climatological variables and occupation of the hotel. Obtaining a prediction model, based on data mining techniques, provides to energy specialists a support tool in decision making.

The KDD data mining methodology was the one selected to develop the experimentation and the most appropriate data mining techniques turned out to be: ANN, SVM and DT.

The experimentation is developed in Weka and MATLAB, it consists in trying different variants for each technique, modifying architecture and training parameters.

They were experimented with 225 variants of the MLP, 90 variants of the FFNN, 45 variants of the LRN, 45 variants of the GRNN, 45 variants of the NARX, 45 variants of the DT and 45 variants of the SVM.

Two models are proposed: for high season the DT (created, trained and tested with a seed of random generation equal to 3 and the dataset of high season without selection of attributes) with an effectiveness of 93.99%; and for low season the DT (created, trained and tested with a seed of random generation equal to 1 and the low season dataset without selection of attributes) with an effectiveness of 96.61%. To interact with these models, a graphical interface was developed in MATLAB.

**Keywords:** model, prediction, electric power, building.

# Contenido

Introducción.....	1
Capítulo I: Marco teórico de la investigación .....	7
1.1. Características del consumo de la electricidad en las edificaciones hoteleras ...	7
1.2. Principales características del consumo de energía eléctrica en Cuba.....	7
1.2.1 Marco regulatorio actual del país en relación a la eficiencia energética [40]	9
1.3. Principales características del consumo de energía eléctrica en los hoteles cubanos .....	9
1.3.1 Variables que influyen en el consumo de energía eléctrica de los hoteles .	10
1.4. Introducción a la inteligencia artificial y a la minería de datos.....	11
1.5. Metodologías de Minería de Datos.....	12
1.5.1 KDD .....	13
1.5.2 CRISP-DM.....	15
1.5.3 SEMMA .....	15
1.6. Diferentes técnicas dentro de la minería de datos .....	16
1.6.1 Árboles de decisión (Decision Tree; DT) .....	16
1.6.2 Máquina de Soporte Vectorial (Support Vector Machine; SVM) .....	17
1.6.3 Redes Neuronales Artificiales.....	17
1.7. Técnicas de Inteligencia Artificial aplicadas a la predicción del consumo de energía eléctrica en edificaciones hoteleras. Descripción de modelos existentes .....	22
Conclusiones del capítulo .....	23
Capítulo II: Diseño del experimento .....	24
2.1 Descripción de las herramientas utilizadas para el desarrollo del experimento	24
2.1.1 WEKA.....	24
2.1.2 MATLAB .....	25
2.2 Comportamiento de las variables presentes en el problema de investigación ..	26
2.3 Preparación de los datos.....	28
2.4 Generalidades sobre la experimentación en MATLAB con los modelos de minería de datos .....	32
2.4.1 Procedimiento de experimentación.....	33
2.5 Modelos utilizados en la experimentación.....	36

2.5.1	Modelo MLP: Teoría y Experimentación.....	36
2.5.2	Modelo FFNN: Teoría y Experimentación.....	38
2.5.3	Modelo LRN: Teoría y Experimentación .....	39
2.5.4	Modelo GRNN: Teoría y Experimentación.....	40
2.5.5	Modelo NARX: Teoría y Experimentación .....	42
2.5.6	Modelo Árbol de Decisión: Teoría y Experimentación .....	43
2.5.7	Modelo Máquina de Soporte de Vectorial: Teoría y Experimentación .....	43
	Conclusiones del capítulo .....	44
	Capítulo III: Análisis de los resultados.....	46
3.1	Criterio de medida del desempeño de los modelos: RMSE. ....	46
3.2	Selección de la mejor variante de los modelos implementados por cada dataset según el criterio RMSE .....	47
3.2.1	Con el dataset de la temporada alta .....	47
3.2.2	Con el dataset de la temporada baja .....	59
3.2.3	Con el dataset del año completo .....	71
3.3	Mejor modelo por dataset.....	84
3.4	Modelos de minería de datos propuestos para la predicción de energía eléctrica de una edificación hotelera .....	85
3.4.1	Mejores modelos según el criterio RMSE .....	85
3.5	Interfaz gráfica .....	85
	Conclusiones del capítulo .....	86
	Conclusiones.....	88
	Recomendaciones.....	89
	Referencias Bibliográficas .....	90
	Glosario de términos .....	94
	Anexos .....	95

## Introducción

El fenómeno de la electricidad se ha estudiado desde la antigüedad, pero su estudio científico sistemático comenzó en los siglos XVII y XVIII. A finales del siglo XIX, los ingenieros lograron aprovecharla para uso doméstico e industrial. La rápida expansión de la tecnología eléctrica la convirtió en la columna vertebral de la sociedad industrial moderna. (Jones, s. f.)

Actualmente la energía es el alma de las sociedades modernas (Kadir & Nora, 2018). La electricidad tiene un sinnúmero de aplicaciones tanto para uso doméstico, industrial, medicinal y en el transporte. Pero este uso ha provocado que las demandas de energía eléctrica sean cada vez mayores y más difíciles de satisfacer por los gobiernos mundiales. El uso de combustibles fósiles para la obtención de la misma ha dado al traste con una contaminación ambiental nunca antes vista en la historia de este planeta. Por tal razón, en todos los países se realizan estrategias para disminuir el consumo de energía eléctrica.

Ejemplo de lo anterior lo constituye la Estrategia Energética en la Unión Europea (Marín Quemada, 2012) iniciada en 2010 con una duración prevista de 10 años. Dentro de la misma podemos contemplar como algunos de sus objetivos son alcanzar el 20% de fuentes renovables en el consumo energético de la Unión Europea en 2020 y un 10% en el sector del transporte, y aumentar la eficiencia energética con el fin de ahorrar un 20% del consumo energético de la Unión Europea respecto de las proyecciones para el año 2020. (GRUPO RENDER INDUSTRIAL S.L, 2015)

Por otra parte, en el continente africano se está librando una lucha a favor de la energía solar como fuente alternativa a la crisis energética en tal continente. Programas como Lighting Africa («Lighting Africa», 2018), han apostado a favor de proyectos de desarrollo de los sistemas de energía solar para viviendas en el continente africano con resultados favorables para su población. (Attanasio & Giorgi, s. f.) Es reconocible también la labor de la región latinoamericana en el uso eficiente de la energía. Ejemplo de lo anterior es la construcción de parques eólicos en casi todos los países y las hidroeléctricas en Brasil, Colombia y Paraguay, principalmente.

En países como Ecuador se han impulsado programas para disminuir los costos y consumos de electricidad como el cambio de focos incandescentes por ahorradores y de refrigeradores de alta demanda de energía, así como el mantenimiento a las redes de alumbrado público (de Armas, s. f.). Los Gobiernos de Perú, Venezuela y Panamá, decretaron leyes para reglamentar la eficiencia energética, mientras que Guatemala, El Salvador, Nicaragua y República Dominicana elaboraron proyectos de ley en el área. Uruguay publicó en 2009 leyes vinculadas a la eficiencia energética, así como un estatuto que conforma un fideicomiso para el financiamiento de proyectos en eficiencia energética. Por su parte, Chile cuenta con la Agencia de Eficiencia Energética («Sitio oficial de la Agencia Chilena de Eficiencia Energética», 2019), institución especializada en el tema. Costa Rica cuenta con la ley de Uso Racional y Eficiente de Energía (UREE) («Proyecto reforma la ley 7447. UREE en Costa Rica», 2019), la más antigua de la región, vigente desde 1994, seguida por las UREE de Brasil y Colombia (El Congreso de Colombia, 2001) en 2001. («América Latina es líder en el ahorro energético mundial», s. f.)

En mayo del 2004 surge en Cuba, como iniciativa del Comandante en Jefe Fidel Castro, la llamada Revolución Energética (Sánchez Borroto, Melo Espinosa, Piloto Rodríguez, & Sierens, 2014). El objetivo fundamental de este proceso era transformar radicalmente el proceso de generación y ahorro de electricidad, el cual se inició aceleradamente en el 2005 y pronto se tradujo en bienestar y calidad de vida para la población. Lo que comenzó como solución a un problema crítico se ha convertido en una estrategia de empleo racional de la energía («Revolución Energética en Cuba», 2015). Es en este marco donde múltiples investigaciones se han desarrollado con vistas a reducir el consumo eléctrico.

El desarrollo de investigaciones orientadas a la predicción del consumo de energía eléctrica en edificios ha aumentado considerablemente. De esta manera han sobresalido autores como Zhao y Magoulès (Hai-xiang & Frédéric, 2012), Fumo (Fumo, 2013), Li y Wen (Li & Wen, 2014), Li Z, Han Y, Xu P. (Zhengwei, Yanmin, & Peng, 2014), Chalal ML, Benachir M, White M y Shrahily R. (Chalal, Benachir, White, & Shrahily, 2016), Wang y Srinivasan (Z. & RS, 2016), etc., los cuales han hecho uso de las técnicas de Inteligencia Artificial (Artificial Intelligence; AI) (Khaled Alsedrah, 2017) para lograr tales objetivos.

En Cuba se han desarrollado una serie de investigaciones relacionadas a la predicción del consumo de energía eléctrica. Ejemplo de lo anterior es el trabajo “Modelo matemático

para estimar el consumo de energía eléctrica en instalaciones de la educación superior”, el cual presenta una alternativa de predicción de energía eléctrica mediante el uso de un modelo estadístico de correlación y regresión lineal múltiple (Sicilia Camacho, Herrera Prat, García Fernández, & Espinosa Nápoles, 2011). La investigación titulada “Pronóstico de consumo de energía eléctrica usando redes neuronales artificiales” desarrollada por un conjunto de profesores de la Universidad de Las Tunas, propone el uso de las Redes Neuronales Artificiales (Artificial Neural Networks; ANN) (Kuldeep, 2016) por ser, según su estudio, el método de pronóstico más preciso (Santiesteban Velázquez, Osvaldo Nápoles González, Rodríguez Peña, & Jaime García, 2014). Otro ejemplo es el trabajo titulado “Modelo para la Predicción Energética de una Instalación Hotelera”, el cual se desarrolló en el hotel Meliá Habana de la Ciudad de la Habana («Sitio oficial del Hotel Meliá Habana de la Ciudad de la Habana», 2019), y que propone el uso del método de series de tiempo radiantes (Radiant Time Series; RTS) (V. Acosta, 2011).

Tomando en cuenta lo anterior se concluye que es prioridad del gobierno de Cuba el no escatimar esfuerzos en pos de lograr un uso eficiente de la energía eléctrica. Es por esto que se propone en esta ocasión el uso de las técnicas de inteligencia artificial para la predicción del consumo de energía eléctrica. Esta predicción hace posible conocer de antemano posibles sucesos o circunstancias a suceder, lo cual a su vez brinda la potencialidad de poder tomar medidas en pos de corregir errores y encausar el futuro; así como planificar o cancelar eventos o actividades, teniendo en cuenta los resultados de tal predicción.

El turismo es el responsable del 5 % de las emisiones globales de dióxido de carbono. Con las tecnologías actuales estas emisiones pudieran aumentar el doble. El sector está siendo monitorizado por altos consumos energéticos y su aporte al calentamiento global (V. Acosta, 2011). Los sistemas que consumen más energía eléctrica dentro de los hoteles son: climatización (42%) y alumbrado (36%), mientras que los motores, elevadores, refrigeración y servicios de lavandería cada uno consume entre un 5% y un 7% de energía. En cambio, para hoteles del Caribe el consumo de climatización es un tanto mayor, en el orden del 55% al 65%, debido fundamentalmente a las altas temperaturas ambientales, mientras que la parte de refrigeración consume un 14%,

alumbrado un 11%, ventiladores y bombas el 12% y la producción de agua caliente de un 7%.

Varios autores consideran que una instalación hotelera para que funcione eficientemente, desde el punto de vista energético, necesita menos del 5% de sus costos con respecto a los ingresos para cubrirlo. En Cuba los hoteles de las cadenas CUBANACAN («Sitio oficial de la cadena hotelera CUBANACAN», 2019), GRAN CARIBE («Sitio oficial de la cadena hotelera GRAN CARIBE», 2019), HORIZONTES («Cadena hotelera HORIZONTES», s. f.) este indicador oscilan entre el 8% y el 16 % pudiendo llegar hasta el 20% en hoteles con una infraestructura muy atrasada de su equipamiento tecnológico y bajos niveles de comercialización. (Cabrera Gorrin & Pérez Dorta, 2019)

Por todas las razones anteriormente expuestas urge una solución al siguiente **problema científico**: ¿Cómo predecir el consumo de energía eléctrica de una edificación hotelera a partir de variables climatológicas y de ocupación?

Se identifica como **objeto de estudio**: los procesos de predicción del consumo de energía eléctrica de edificaciones hoteleras en climas tropicales. Se define como **campo de acción**: la aplicación de técnicas de minería de datos a la predicción del consumo de energía eléctrica en edificaciones hoteleras.

### **Idea a Defender**

Con la obtención de un modelo, basado en técnicas de minería de datos, para predecir el consumo de energía eléctrica de una edificación hotelera, será posible brindar a los especialistas en energía una herramienta de apoyo en la toma de decisiones.

La labor científico investigativa está encaminada a dar cumplimiento al siguiente **objetivo general**: Proponer un modelo, basado en técnicas de minería de datos, para predecir el consumo de energía eléctrica de una edificación hotelera a partir de variables climatológicas y de ocupación.

### **Objetivos específicos**

- 1- Seleccionar la metodología de minería de datos más apropiada para el tratamiento de los datos del problema.
- 2- Seleccionar las técnicas de minería de datos apropiadas.
- 3- Seleccionar los algoritmos con mejores resultados en la predicción del consumo de energía eléctrica en edificaciones hoteleras.
- 4- Experimentar en MATLAB con variantes de los algoritmos seleccionados.
- 5- Comparar los resultados obtenidos para proponer el modelo de predicción que mejor se ajusta al problema.

### **Aporte práctico**

Los aportes prácticos de la presente investigación son: La obtención del modelo más eficiente para predecir el consumo diario de energía eléctrica de una edificación hotelera, a partir de las variables climatológicas y de ocupación del hotel; y la implementación de una interfaz gráfica en MATLAB que permita al cliente interactuar con el modelo.

El trabajo se estructura de la siguiente forma: introducción, tres capítulos, conclusiones, bibliografía y anexos.

**Capítulo 1** nombrado “Marco teórico de la investigación”. Se exponen los conceptos asociados al dominio del problema, el uso de la energía eléctrica en edificaciones hoteleras, varios modelos de predicción encontrados en la literatura, así como las ventajas de utilizar una metodología de minería de datos para encontrar modelos eficientes.

**Capítulo 2** nombrado “Diseño del experimento”: Se presentan las características (arquitectura y aprendizaje) generales de los modelos de ANN seleccionados y los detalles de su experimentación en Weka y MATLAB.

**Capítulo 3** nombrado “Análisis de los resultados”: Se comparan los resultados obtenidos por las diferentes variantes de los modelos utilizados para llegar a conclusiones respecto al mejor en la predicción del consumo de energía eléctrica de edificaciones hoteleras.

Para el desarrollo de esta investigación se emplearon diferentes métodos teóricos de investigación que ayudaron a fundamentar, con el rigor científico requerido, la propuesta de caso de estudio.

- Método Histórico y lógico: Facilitó el análisis de los antecedentes históricos de las investigaciones desarrolladas para la predicción del consumo de energía eléctrica, en el mundo y particularmente en Cuba y Cienfuegos.
- Método Análisis y síntesis: Ayudó en la valoración y sistematización del estado del arte sobre el tema a nivel internacional y regional, facilitando la comprensión y explicación de su evolución. Los resultados de la aplicación de este método identificaron las principales tendencias relacionadas con el caso de estudio. Se utiliza además para el análisis de la bibliografía consultada, así como en la interpretación de las diferentes etapas investigativas con el fin de organizar y estructurar la investigación. Estará presente en toda la investigación y en todas las tareas.
- Método Inducción-deducción: Permite el estudio de los modelos de predicción de energía eléctrica, a partir de la necesaria articulación filosófica de lo general, lo particular y lo singular y de los postulados teóricos generales que promueven el desarrollo de una organización del conocimiento mediante sistemas curriculares. Este método facilitó la valoración de los estudios desarrollados sobre el tema, así como la importancia y alcance de su aplicación en una edificación hotelera.
- Modelación: Su uso permite hacer una representación teórica, metodológica y práctica del modelo propuesto. Y las herramientas informáticas que se utilizan son Weka, para la preparación de los datos y MATLAB para el desarrollo de los modelos de predicción.

El método empírico de la investigación utilizado es la entrevista, la cual se utilizará como una conversación planificada con el cliente, para obtener información acerca del problema en cuestión. Su uso constituye un medio para el conocimiento cualitativo de las características particulares del proceso investigado y puede influir en el desarrollo del modelo de predicción de energía eléctrica deseado.

## **Capítulo I: Marco teórico de la investigación**

El presente capítulo enuncia las principales características vinculadas al objeto de estudio, deja claro ciertos aspectos que inciden de manera directa sobre el consumo de energía eléctrica en edificaciones hoteleras. Se analizan diferentes metodologías de minería de datos y se escoge la que será empleada en el desarrollo de la presente investigación. Y, por último, se analizan las técnicas de minería de datos que son más utilizadas dentro del contexto de la presente investigación.

### **1.1. Características del consumo de la electricidad en las edificaciones hoteleras**

Según las estadísticas de la Agencia Internacional de Energía («Sitio oficial de la Agencia Internacional de Energía», 2019) los edificios son responsables del 32% del consumo total final de energía, incrementándose en zonas orientadas a la industria. Ejemplo de esto lo constituye Hong Kong donde los edificios consumen más del 90% de la electricidad (Cheng, Fu, & Shengwei, 2014). Específicamente en la Unión Europea consumen más del 40% (Li & Wen, 2014).

En el caso de los Estados Unidos el consumo anual total del país entre los años 1996 y 2006 se considera como el 1.9% del consumo mundial anual en estos años, mientras que para el 2030 se estima que alcanzará el 26%. Más del 70% de este consumo es provocado por las edificaciones (Li & Wen, 2014).

### **1.2 Principales características del consumo de energía eléctrica en Cuba**

De acuerdo a estadísticas del Ministerio de la Industria Básica («Sitio oficial del Ministerio de la Industria Básica de Cuba», 2019), hasta los años 2004 y 2005, la generación eléctrica se basaba en 11 grandes e ineficientes plantas termoeléctricas, con un promedio de 25 años de explotación, sólo 60% de disponibilidad, frecuentes averías y altos

insumos. Hasta 2006, se sufrían frecuentes apagones, de hasta 20 horas diarias en ciertos casos, y principalmente en horarios de máxima demanda.

En medio de esta crítica situación, se definió una política energética nacional, basada en programas de acción. En primer lugar, se llevó adelante el programa de uso racional de la energía, con el objetivo de lograr ahorro y uso eficiente. En segundo término, se implementó el programa de transformaciones del sistema eléctrico nacional, que se propuso el incremento de la disponibilidad del servicio eléctrico. Se inició asimismo el programa para incrementar el uso de las energías renovables, el programa de aumento de la explotación y producción de petróleo y gas, y el de auspicio a la colaboración internacional.

La política energética cubana ha tenido resultados auspiciosos, considerando que en 2004 y 2005 se enfrentaron 188 y 224 días con apagones mayores a 100 MW<sup>1</sup> de más de una hora de duración, mientras que en 2006 sólo fueron 3 días, y desde 2007 éstos han sido superados. De acuerdo a la Unión Eléctrica de Cuba («Sitio oficial de la Unión Eléctrica de Cuba», 2018), el país ahorró en 2006 y 2007 aproximadamente 960 mil toneladas de combustibles, mientras que el Ministerio de la Industria Básica calcula en 400 millones de dólares el monto ahorrado, como resultado de las medidas adoptadas a través de los diferentes programas de la política energética. (Domingo Laino, 2015)

Actualmente el 99 % de la población cubana tiene acceso al servicio eléctrico, proporción que supera el promedio actual del área latinoamericana y del Caribe en más de un 20%. El sistema brinda servicios a toda la esfera productiva del país y cuenta actualmente con una capacidad de generación instalada superior a los 3000 MW, capaz de satisfacer una demanda máxima anual superior a los 2000 MW. El consumo anual de petróleo combustible en el país asciende a unos 6 millones de toneladas y de ellos el 55 % se destina a la generación de electricidad en termoeléctricas.

---

<sup>1</sup> Ver MW en el glosario de términos

### 1.2.1 Marco regulatorio actual del país en relación a la eficiencia energética (ONURE, 2017)

- **Política para el desarrollo y sostenimiento de las Fuentes Renovables de Energía (FRE) y el incremento de la Eficiencia y Conservación Energética.** Establece acciones concretas para elevar la participación de las FRE en la generación de electricidad hasta alcanzar una proporción de no menos del 24 % hasta el 2030.
- **Decreto 327 para las Inversiones. y Res 283/2014 del MINEM** («Sitio oficial del Ministerio de Energía y Minas de Cuba», 2019). Requisitos de eficiencia, conservación energética y aprovechamiento de las fuentes de energías renovables para la aprobación de la Licencia Energética en el proceso inversionista.
- **Resolución 136/2009** del hoy MINEM “Reglamento Técnico de Eficiencia Energética para los Equipos de Uso Final de la Energía Eléctrica”
- **Resolución No. 316/2008 del MICONS** («Sitio oficial del Ministerio de la Construcción de Cuba», 2019). Mandata el uso obligatorio, para todas las inversiones constructivas, de la NC 220 “Edificaciones. Requisitos de diseño para la eficiencia energética”
- **Resolución 328/2007** Establece el plan de consumo y se crea grupos de Fiscalización y Supervisión de la UNE y CUPET («2019», s. f.-a).
- **Orden Ministerial - 863** Implementación del Manual de portadores energéticos.
- **Decreto 345/2017** del Consejo de Ministros “Del desarrollo de las Fuentes Renovables de Energía y Eficiencia Energética”.

### 1.3 Principales características del consumo de energía eléctrica en los hoteles cubanos

En la actualidad, Cuba se encuentra entre los países donde el turismo figura en el plan de desarrollo como la fuente principal de divisas. A la par del crecimiento del sector turístico, tanto hotelero como extra hotelero, este ha debutado como una de las empresas más consumidoras de energía eléctrica en el país (Molina González, Velarde Bedregal, E. Santiesteban Toca, & P. Monteagudo Yanes, 2017). La actividad turística, por su

propia definición, supone que un visitante ocupe un espacio geográfico distinto de aquél en el que tiene su residencia habitual durante un período de tiempo limitado.

Para posibilitar y potenciar el uso con fines turísticos de los recursos naturales de un territorio, el hombre debe desarrollar una infraestructura que permita acoger a los visitantes, de modo que se adecue el territorio para su uso como destino turístico; es decir, se construya un espacio turístico. El costo de la producción o prestación de servicios de su hotel puede definirse como la expresión monetaria de los recursos de todo tipo empleados en el proceso de atención a los huéspedes y usuarios institución. Las condiciones de competitividad demandan una mayor eficiencia en la operación de la instalación turística de los hoteles.

### **1.3.1 Variables que influyen en el consumo de energía eléctrica de los hoteles**

Es importante conocer las variables que influyen en el consumo de energía eléctrica de los hoteles para de esa forma tratar de abatir el impacto de ellas sobre el consumo de energía eléctrica. Algunas de estas variables son:

**Ocupación:** Una de las variables que más afectan el consumo de energía de un hotel es su estado ocupacional. Lógicamente al incrementarse la cantidad de usuarios dentro de la instalación hotelera se incrementará el número de equipos eléctricos necesarios para el disfrute de dichos usuarios, incidiendo de manera directa sobre el consumo total de energía eléctrica de la edificación.

**El clima:** Esta variable es la más importante en el consumo de energía eléctrica y en los países del trópico en ocasiones se puede consumir en una misma habitación hasta 10 veces más energía en verano, comparándolo con el consumo de invierno. Ella está muy relacionada en el caso cubano con la época del año donde los meses de julio y agosto son los de mayor calor del país, y meses como mayo-junio, septiembre-octubre las temperaturas promedio son inferiores debido al efecto del incremento de la lluvia y con ello ocurre un refrescamiento de las temperaturas exteriores. (Cabrera Gorrin & Pérez Dorta, 2019)

**Época del año:** Se acostumbra hablar de época de verano como aquella con un mayor periodo de calor en los meses de julio y agosto. Sin embargo, el país, por su ubicación geográfica presenta altas temperaturas aún en los meses llamados invernales. Esta variable está estrechamente relacionada con la primera, por lo que su incidencia en el consumo de energía eléctrica es muy similar. (Ramos Niembro, Fiscal Escalante, Maqueda Zamora, Sada Gámiz, & Buitrón Sánchez, 2016)

**Forma de uso de los equipos o hábitos:** Dependiendo de la manera en que las personas estén acostumbradas a usar los equipos es como van estos equipos a consumir electricidad, produciendo una no despreciable diferencia en tal consumo si se comparan los usos o costumbres de personas de diferentes regiones del mundo, o incluso, del país. (Ramos Niembro et al., 2016)

## **1.4 Introducción a la inteligencia artificial y a la minería de datos**

Existen muchas definiciones de la AI, las cuales hacen énfasis en diferentes aspectos. Una definición clásica sostiene que “La Inteligencia Artificial es la parte de las Ciencias de la computación que se ocupa del diseño de sistemas de computación inteligentes, esto es, sistemas que exhiben las características que asociamos con la inteligencia en el comportamiento humano”. Otra definición es “La Inteligencia Artificial es el estudio de cómo hacer que las computadoras hagan cosas que, hasta el momento, los humanos hacen mejor”. Esta definición muestra el aspecto evolutivo de este campo de investigación y desarrollo, el cual experimenta rápidos cambios de foco y alcance. (Casali, s. f.)

El aumento del volumen y variedad de información que se encuentra informatizada en bases de datos digitales y otras fuentes ha crecido espectacularmente en las últimas décadas. Además, ya que los datos pueden proceder de fuentes diversas y pertenecer a diferentes dominios, parece clara la inminente necesidad de analizar los mismos para la obtención de información útil para la organización.

En muchas situaciones, el método tradicional de convertir los datos en conocimiento consiste en un análisis e interpretación realizada de forma manual. Esta forma de actuar

es lenta, cara y altamente subjetiva. De hecho, el análisis manual es impracticable en dominios donde el volumen de los datos crece exponencialmente: la enorme abundancia de datos desborda la capacidad humana de comprenderlo sin la ayuda de herramientas potentes. Consecuentemente, muchas decisiones importantes se realizan, no sobre la base de la gran cantidad de datos disponibles, sino siguiendo la propia intuición del usuario al no disponer de las herramientas necesarias.

Todos estos problemas y limitaciones de las aproximaciones clásicas han hecho surgir la necesidad de una nueva generación de herramientas y técnicas para soportar la extracción de conocimiento útil desde la información disponible, y que se engloban bajo la denominación de minería de datos. (Hernández Orallo, Ramírez Quintana, & Ferri Ramírez, 2011) Entonces, ¿qué es minería de datos?

Según (Camejo Corona, 2012) la minería de datos (Data Mining; DM) consiste en la extracción no trivial de información que reside de manera implícita en los datos. Dicha información era previamente desconocida y podrá resultar útil para algún proceso. En otras palabras, la minería de datos prepara, sondea y explora los datos para sacar la información oculta en ellos. De una manera simplista pero ambiciosa, podríamos decir que el objetivo de la minería de datos es convertir datos en conocimiento, lo cual no es sólo ambicioso sino muy amplio (Hernández Orallo et al., 2011).

## **1.5 Metodologías de minería de datos**

Los esfuerzos en el área de la minería de datos se han centrado en su gran mayoría en la investigación de técnicas para la explotación de información y extracción de patrones. Sin embargo, se ha profundizado en menor medida el hecho de cómo ejecutar este proceso hasta obtener el “nuevo conocimiento”, es decir, en las metodologías. Las metodologías permiten llevar a cabo el proceso de minería de datos en forma sistemática y no trivial. Ayudan a las organizaciones a entender el proceso de descubrimiento de conocimiento y proveen una guía para la planificación y ejecución de los proyectos. Una metodología no solo define las fases de un proceso sino también las tareas que deberían realizarse y cómo llevar a cabo las mismas. (Toledo, 2017) A continuación se hará referencia a las principales metodologías de minería de datos utilizadas por los analistas para la realización de proyectos de Data Mining.

### 1.5.1 KDD

KDD («Sitio oficial de KDD», 2019) es una metodología genérica para encontrar información en un gran conjunto de datos y con ello generar conocimiento. Se define como un proceso no trivial de extracción de información a partir de los datos, la cual se encuentra presente de forma implícita, previamente desconocida y potencialmente útil para el usuario o para el negocio. El objetivo principal de esta metodología es automatizar el procesamiento de los datos, permitiendo a los usuarios dedicar más tiempo a las tareas de análisis y al descubrimiento de relaciones entre los datos.

El KDD es un proceso que consta de una serie de etapas consecutivas, y funciona de forma iterativa e interactiva (León Guzmán, 2012). Iterativa, ya que es posible regresar desde cualquier etapa a una anterior para ajustar los parámetros o supuestos previos, e interactiva pues el usuario experto del negocio tiene que estar presente para aportar con su conocimiento en la preparación de los datos y en la validación de los resultados que se obtengan durante el proceso (ROQUE MONTALVO, 2016).

**A continuación se muestran cada uno de los pasos contenidos en las fases de la metodología de minería de datos KDD (León Guzmán, 2012):**

- **Formulación del problema:** No todo es matemática y estadística, sino entender la problemática a la que se va a enfrentar y tener contexto para proponer soluciones viables y reales. Es importante conocer las propiedades, limitaciones y reglas del escenario en estudio, para posteriormente definir las metas a alcanzar.
- **Recopilación de datos:** Del conjunto de datos recolectados y ya definidos los objetivos por alcanzar, se deben elegir datos disponibles para realizar el estudio e integrarlos en uno solo que puedan favorecer a llegar a alcanzar a los objetivos del análisis. Muchas veces esta información puede encontrarse en una misma fuente (centralizado) o pueden estar distribuidos.
- **Pre-procesamiento – limpieza:** En esta etapa se determina la confiabilidad de la información, es decir, realizar tareas que garanticen la utilidad de los datos. Para esto se hace la limpieza de datos (tratamiento de datos perdidos o remover valores atípicos). Esto implica eliminar variables o atributos con datos faltantes o eliminar información no útil.

- **Transformación:** En esta etapa se mejora la calidad de los datos con transformaciones que involucren ya sea reducción de dimensionalidad (disminuir la cantidad de variables del conjunto de datos) o bien transformaciones como por ejemplo convertir los valores que son números a categóricos (discretización).
- **Elegir la tarea de minería:** Fase en la que se refiere a elegir el paradigma apropiado de Minería de Datos, ya sea la clasificación, regresión o agrupación, según los objetivos que se haya planteado para la investigación (predicción o descripción), la primera ocupada para encontrar un modelo que sea utilizada para casos futuros y desconocidos; mientras que la segunda solo para observar su comportamiento.
- **El método de minería:** Posteriormente se procede a seleccionar la técnica o algoritmo, o incluso más de uno para la búsqueda del patrón y obtener conocimiento. El meta-aprendizaje se enfoca en explicar la razón por la que un algoritmo funciona mejor en determinadas problemáticas, y para cada técnica existen diferentes posibilidades de cómo seleccionarlas. Cada algoritmo tiene su propia esencia, su propia manera de trabajar y obtener los resultados, por lo que es recomendable conocer las propiedades de aquellos candidatos a utilizar y ver cual se ajusta mejor a los datos.
- **Aplicación del algoritmo:** Por fin, una vez seleccionado las técnicas el paso siguiente es aplicarlo a los datos ya seleccionados, limpiados y procesados. Es posible que la ejecución de los algoritmos sean varias intentando ajustar los parámetros que optimicen los resultados. Estos parámetros varían de acuerdo al método seleccionado.
- **Evaluación de resultados:** Una vez aplicado los algoritmos al conjunto de datos, se procede a evaluar los patrones que se generaron y el rendimiento que se obtuvo para verificar que cumpla con las metas planteadas en las primeras fases.
- **Visualización:** Si todos los pasos se siguen correctamente y los resultados de la evaluación son satisfactorios, la última etapa es simplemente aplicar el conocimiento encontrado al contexto y comenzar a resolver sus problemáticas. Si de lo contrario, los resultados no son satisfactorios entonces es necesario regresar a las anteriores etapas a realizar algún ajuste, analizando desde la selección de los datos hasta en la etapa de evaluación.

## 1.5.2 CRISP-DM

La metodología CRISP-DM («CRISP-DM», s. f.) consta de cuatro niveles de abstracción, organizados de forma jerárquica en tareas que van desde el nivel más general hasta los casos más específicos. El proceso está organizado en fases, estando cada fase a su vez estructurada en varias tareas generales de segundo nivel. Las tareas generales se proyectan a tareas específicas, donde se describen las acciones que deben ser desarrolladas para situaciones específicas. El cuarto nivel, recoge el conjunto de acciones, decisiones y resultados sobre el proyecto de data mining específico. (Toledo, 2017)

## 1.5.3 SEMMA

SEMMA («SEMMA», 2015), definida como el proceso de selección, exploración y modelado de grandes cantidades de datos para descubrir patrones de negocio desconocidos. Su nombre es el acrónimo correspondiente a las cinco fases básicas del proceso: Sample (Muestreo), Explore (Exploración), Modify (Manipulación), Model (Modelado), Assess (Valoración).

El proceso se inicia con la extracción de la población muestral sobre la que se va a aplicar el análisis. El objetivo de esta fase consiste en seleccionar una muestra representativa del problema en estudio. Una vez determinada esta muestra se procede a una exploración de la información disponible con el fin de simplificar el problema para optimizar la eficiencia del modelo. Para lograr este objetivo se propone la utilización de herramientas de visualización o de técnicas estadísticas para así determinar cuáles son las variables explicativas que van a servir como entradas al modelo. La tercera fase consiste en la manipulación de los datos, en base a la exploración realizada, de forma que se definan y tengan el formato adecuado los datos que serán introducidos en el modelo. La siguiente fase es el análisis y modelado de los datos, que consiste en establecer una relación entre las variables explicativas y las variables objeto del estudio, que posibiliten inferir el valor de las mismas con un nivel de confianza determinado. Finalmente, la última fase del proceso consiste en la valoración de los resultados mediante el análisis de bondad del modelo o modelos, contrastado con otros métodos estadísticos o con nuevas poblaciones muestrales. (Toledo, 2017)

## KDD: Metodología de Minería de datos seleccionada para el desarrollo de esta investigación

La selección se justifica en que el problema a resolver es supervisado de no mucha complejidad que precisa de una metodología ágil y no robusta como lo es CRISP-DM. Además, las etapas que describe KDD como metodología se ajustan perfectamente al problema en cuestión, abarcando todos sus límites y restricciones.

### 1.6 Diferentes técnicas dentro de la minería de datos

#### 1.6.1 Árboles de decisión (Decision Tree; DT)

De todos los métodos de aprendizaje, los sistemas de aprendizaje basados en árboles de decisión (Lior & Oded, 2015) son quizás el método más fácil de utilizar y de entender. Un árbol de decisión es un conjunto de condiciones organizadas en una estructura jerárquica, de tal manera que la decisión final a tomar se puede determinar siguiendo las condiciones que se cumplen desde la raíz del árbol hasta alguna de sus hojas. Una de las grandes ventajas de los árboles de decisión es que, en su forma más general, las opciones posibles a partir de una determinada condición son excluyentes. Esto permite analizar una situación y, siguiendo el árbol de decisión apropiadamente, llegar a una sola acción o decisión a tomar. Tomando en cuenta lo anterior se puede concluir que: un árbol de decisión está formado por un conjunto de nodos de decisión (interiores) y de nodos-respuesta (hojas):

- Un **nodo de decisión** está asociado a uno de los atributos y tiene 2 o más ramas que salen de él, cada una de ellas representando los posibles valores que puede tomar el atributo asociado. De alguna forma, un nodo de decisión es como una pregunta que se le hace al ejemplo analizado, y dependiendo de la respuesta que dé, el flujo tomará una de las ramas salientes.
- Un **nodo-respuesta** está asociado a la clasificación que se quiere proporcionar, y devuelve la decisión del árbol con respecto al ejemplo de entrada. (Sancho Caparrini, s. f.)

### **1.6.2 Máquina de Soporte Vectorial (Support Vector Machine; SVM)**

La teoría de las Máquinas de Soporte Vectorial (Carmona Suárez, 2013) fue desarrollada por Vapnik («Vladimir Vapnik», 2019) basado en la idea de minimización del riesgo estructural (SRM, por sus siglas en inglés). A diferencia de las Redes Neuronales Artificiales que utilizan durante la fase de entrenamiento, el principio de Minimización del Riesgo Empírico (ERM, por sus siglas en inglés), las SVM se basan en el principio de Minimización del Riesgo Estructural, la cual ha mostrado un mejor desempeño que el ERM, ya que las Máquinas de Soporte Vectorial minimizan un límite superior al riesgo esperado a diferencia del ERM que minimiza el error sobre los datos de entrenamiento. Esto, además del uso de una función núcleo o kernel, le atribuye a la SVM una gran capacidad de generalización, incluso cuando el conjunto de entrenamiento es pequeño, lo cual la han puesto en ventaja respecto a otras técnicas populares de clasificación y/o regresión, como las mencionadas ANN.

Las SVM han sido desarrolladas como una técnica robusta para clasificación y regresión aplicado a grandes conjuntos de datos complejos con ruido; es decir, con variables inherentes al modelo que para otras técnicas aumentan la posibilidad de error en los resultados pues resultan difíciles de cuantificar y observar. Se dice que tanto la capacidad de generalización como el proceso de entrenamiento de la SVM, no dependen necesariamente del número de atributos, lo que permite un excelente comportamiento en problemas de alta dimensionalidad.

Las SVM mapean los puntos de entrada a un espacio de características de una dimensión mayor, para luego encontrar el hiperplano que los separe y maximice el margen entre las clases. (Colmenares, 2009)

### **1.6.3 Redes Neuronales Artificiales**

Las redes de neuronas artificiales (denominadas habitualmente como RNA) son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el sistema nervioso biológico. Se trata de un sistema de interconexión de neuronas que colaboran entre sí para producir un estímulo de salida. En inteligencia artificial es frecuente referirse a ellas como redes de neuronas o redes neuronales.

Al igual que el cerebro, una ANN se compone de un conjunto masivamente paralelo de unidades de proceso muy simples llamadas neuronas, y es en las conexiones entre estas unidades donde reside la inteligencia de la red. Cada neurona recibe una serie de entradas a través de interconexiones y emite una salida. En términos de escala, un cerebro es muchísimo mayor que cualquier ANN creada hasta la actualidad, y las neuronas artificiales también son más simples que su contrapartida humana.

Tomando en cuenta lo anterior es bueno añadir algunas de las ventajas del uso de las ANN, las cuales son:

- **Aprendizaje Adaptativo:** Las ANN aprenden a realizar tareas a partir de un conjunto de datos dados, en el proceso de aprendizaje, estos datos son representados como las entradas y pesos.
- **Auto-organización:** Pueden crear su propia organización o representación de la información recibida.
- **Operación en tiempo real:** Las operaciones realizadas pueden ser llevadas a cabo por computadores paralelos, o dispositivos de hardware especiales que aprovechan esta capacidad.
- **Tolerancia a fallos parciales:** La destrucción parcial de una red, daña el funcionamiento de la misma, pero no la destruye completamente. Esto es debido a la redundancia de la información contenida. (Varela Arregocés & Campbells Sánchez, 2011)

### **Algunos consejos tomados de escritores referentes en la inteligencia artificial**

Los problemas de predicción, consisten en hallar la estimación de una variable continua de salida, a partir de la presentación de un conjunto de variables predictivas de entradas (discretas y/o continuas).

Las variables deben seguir una distribución normal o uniforme, y el rango de posibles valores debe ser aproximadamente el mismo y acotado dentro del intervalo de trabajo de la función de activación empleada en las capas ocultas y de salida de la red neuronal. La cantidad de neuronas de entradas es equivalente a la cantidad de variables

independientes que existan en el problema y la cantidad de neuronas en la capa de salida es equivalente a la cantidad de variables dependientes que existan en el problema.

### **Capas Ocultas**

Varios investigadores como: Demuth, Beale y Hagan, (H., M., & M., 2017), (H., M., & M., 2007), (Demuth, Beale, & Hagan, 2010) Tanco (F., 2010), Federico (L., 2005) y Marín (J., 2012), enfatizan en que una red neuronal es eficiente con hasta dos capas ocultas en su arquitectura. Raramente se han desarrollado modelos de ANN con más de dos capas ocultas.

La selección de la cantidad de neuronas en las capas ocultas es un área en la que aún queda mucho por indagar. Por tanto, dicha selección queda en manos de la experiencia de los investigadores y es depende en gran medida de la experimentación. Algunas investigaciones precedentes relacionadas con el tema en estudio, tales como Montelier (S., M., A., J., & C., s. f.) y Visanzay (Visanzay, s. f.), muestran como una red de entre 1 y 4 neuronas en la capa de salida, es eficiente con entre 4 y 25 neuronas aproximadamente para las capas ocultas.

### **Funciones de transferencia**

Demuth, Beale y Hagan (H. et al., 2017), (H. et al., 2007), (Demuth et al., 2010), Tanco (F., 2010), Federico (L., 2005), y Marín (J., 2012), recomiendan dos formas básicas para la función de activación o de transferencia, de las neuronas ocultas y/o de salida en una red multicapa: la función sigmoideal, y la función lineal (o identidad). Para aprovechar la capacidad de las redes neuronales de aprender relaciones complejas o no lineales entre variables, se recomienda la utilización de funciones no lineales al menos en las neuronas de la capa oculta. La elección de la función de activación en las neuronas de la capa de salida dependerá del tipo de tarea impuesto. En tareas de predicción o aproximación de una función, generalmente se emplea la función de activación lineal para la capa de salida. En el caso del presente proyecto y estando de acuerdo con lo descrito con anterioridad se dejó el parámetro de la función de transferencia por defecto en MATLAB («2019», s. f.-b), la cual presenta de esta manera una función de activación tangente-sigmoideal para la(s) capa(s) oculta(s) y una función lineal para la capa de salida.

### 1.6.3.1 Perceptrón Multicapa (Multi-Layer Perceptron; MLP)

El perceptrón multicapa con conexiones hacia adelante es una generalización del perceptrón simple. Surge como respuesta a los problemas que tenía dicha red, como por ejemplo, no poder resolver problemas que no fueran linealmente separables. De hecho, algunos autores han demostrado que el perceptrón multicapa es un aproximador universal de cualquier función continua en el espacio  $\mathbb{R}^n$ .<sup>2</sup> (MORENO RODRÍGUEZ, 2009)

### 1.6.3.2 Red Neuronal de Pre-alimentación (Feed-forward Neuronal Network; FFNN)

Las RNA de topología *feedforward* se caracterizan por ejecutar el procesamiento de datos en una sola dirección, para que la información nunca pueda pasar más de una vez a través de una neurona antes de generarse la respuesta de salida. Estas ANN son relativamente fáciles de programar, por lo que son muy populares entre la comunidad científica. (Vásquez López, 2014)

### 1.6.3.3 Red neuronal artificial de capas recurrentes (Layer Recurrent Neural Network; LRN)

Las redes que poseen conexiones entre pesos de la salida de una capa a la entrada de la misma capa o anteriores son conocidas como redes recurrentes. Las redes recurrentes no tienen memoria, es decir, la salida solamente está determinada por las entradas y los pesos. Las capas recurrentes redireccionan previas salidas a entradas. Su salida es determinada por su entrada y sus salidas previas, por lo que se puede asemejar a la memoria a corto plazo de los seres humanos. (Cárdenes Almeida, s. f.)

### 1.6.3.4 Red neuronal regresiva generalizada (Generalized Regression Neural Network; GRNN)

El modelo GRNN es también conocido como el estimador de Nadaraya-Watson (Nadaraya, E. A., 1964), (Watson, G. S., 1964) o estimador de regresión de kernel. GRNN es un modelo no paramétrico y las predicciones son hechas promediando las salidas de la variable objetivo del dataset usado en el entrenamiento, de acuerdo a su distancia de la observación suministrada. (Makridakis, Spiliotis, & Assimakopoulos, 2018)

---

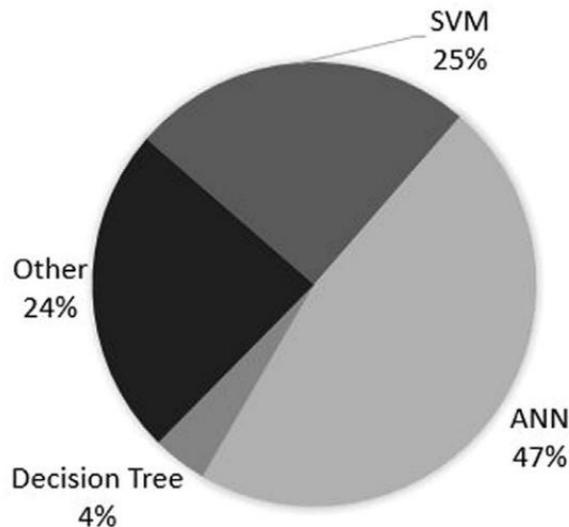
<sup>2</sup> La notación matemática  $\mathbb{R}^n$  se refiere a un espacio de  $n$  dimensiones de los números reales.

### 1.6.3.5 Red Auto Regresiva No Lineal con Aportes Exógenos (Nonlinear Autoregressive Network with Exogenous Inputs; NARX)

Demuth, Beale y Hagan (H. et al., 2007), (Demuth et al., 2010) proponen la red auto regresiva no lineal con aportes exógenos y algunas variantes de esta. La red NARX es una red dinámica parcialmente recurrente, con conexiones de información retroactiva incluyendo varias capas de la red (H. et al., 2007), (Demuth et al., 2010).

#### Técnicas de minería de datos seleccionadas para el desarrollo de la experimentación de la presente investigación

Tomando en consideración la literatura revisada, lo expuesto por Kadir Amasyali y Nora M. El-Gohary (S. et al., s. f.) y la figura 1, las técnicas más utilizadas en la predicción del consumo de energía eléctrica, y por lo tanto escogidas para el desarrollo de la experimentación de la presente investigación, son las ANN, SVM y los DT.



**Fig. 1:** Técnicas de minería de datos más utilizadas en los procesos de predicción de energía eléctrica.

## **1.7 Técnicas de Inteligencia Artificial aplicadas a la predicción del consumo de energía eléctrica en edificaciones hoteleras. Descripción de modelos existentes**

Han sido muchos los modelos de predicción del consumo de energía eléctrica en edificios creados, cada uno con sus características individuales. Una gran cantidad de autores han intervenido en este ámbito exponiendo el uso de las técnicas de la inteligencia artificial. De esta manera destacan investigadores como Zhao y Magoulès (Hai-xiang & Frédéric, 2012), Fumo (Fumo, 2013), Li y Wen (Li & Wen, 2014), Li Z, Han Y, Xu P. (Zhengwei et al., 2014), Chalal ML, Benachir M, White M y Shrahily R. (Chalal et al., 2016), Wang y Srinivasan (Z. & RS, 2016), etc.

Zhao y Magoulès clasificaron métodos de predicción de consumo de energía en edificios en métodos elaborados de ingeniería, métodos simplificados de ingeniería, métodos estadísticos, métodos basados en ANN, métodos basados en SVM, y modelos grises; y guía algunos análisis comparativos en términos de la complejidad modelo, facilidad de uso, velocidad de corrida, entradas necesarias, y la exactitud. Ahmad et al. enfocó la atención en la revisión de métodos basados en ANN, SVM, e híbridos y ha discutido los principios, las ventajas, y las desventajas de estos métodos. (Hai-xiang & Frédéric, 2012)

Fumo resumió la clasificación de los métodos de predicción de consumo de energía en edificaciones propuestos por los estudios diversos y colocó un especial énfasis en la revisión de la calibración del modelo, en la verificación y en los datos del clima usados para modelaje (Fumo, 2013). Li y Wen guiaron una revisión inclusiva; ellos revisaron el estado de los estudios del arte, no solo para el modelamiento y predicción de la energía en edificaciones sino también para el modelamiento de componentes críticos (ej., el modelado de la generación fotovoltaica), modelado energía en edificaciones para dar a respuesta a demandas (ej., el pronóstico del clima), el modelamiento de la energía en edificaciones basada en agentes, y la identificación de sistema para el modelado del consumo de energía eléctrica en edificaciones (Li & Wen, 2014).

Li Z, Han Y, Xu P. revisaron los métodos para el consumo de energía en edificios comparativamente y propusieron un diagrama de flujo que intenta ayudarle a usuarios a escoger el método correcto de predicción (Zhengwei et al., 2014). Chalal ML, Benachir M, White M y Shrahily R. enfocaron la atención en ambas escalas de edificaciones y discutieron los métodos disponibles dentro de cada escala (Chalal et al., 2016). Wang y Srinivasan revisaron y compararon los principios, las aplicaciones, las ventajas, y las desventajas de métodos basados en AI (ej., ANN y SVM) y los métodos del conjunto (Z. & RS, 2016). Como se ha podido observar la mayoría de estos estudios se enfocan particularmente en los métodos/algoritmos de minería de datos.

## **Conclusiones del capítulo**

- La inteligencia artificial ofrece una amplia gama de técnicas efectivas para la predicción del consumo de energía eléctrica.
- La metodología de minería de datos seleccionada para el desarrollo de la experimentación es KDD.
- Las técnicas de minería de datos más usadas para la predicción del consumo de energía eléctrica en edificaciones hoteleras son: ANN, SVM y los DT.

## Capítulo II: Diseño del experimento

En el presente capítulo se realiza una breve descripción de las herramientas utilizadas durante el desarrollo de la experimentación y se abarca el comportamiento de las variables presentes en la investigación. A continuación se realiza la preparación de los datos, y finalmente se realiza el experimento.

El problema científico de la presente investigación plantea el conflicto: ¿Cómo predecir el consumo de energía eléctrica de una edificación hotelera a partir de variables climatológicas y de ocupación? Para dar solución a tal problema científico se busca en la experimentación obtener uno o varios modelos de minería de datos que con el margen de error menor posible sean capaces de predecir el consumo de energía eléctrica de una edificación hotelera. Teniendo en cuenta que tanto, el valor a predecir, como los utilizados durante el proceso de entrenamiento y prueba son continuos; y siguiendo el quinto paso de la metodología de minería de datos KDD; se puede decir que se está tratando con un problema de regresión.

### 2.1 Descripción de las herramientas utilizadas para el desarrollo del experimento

#### 2.1.1 WEKA

La Weka es un ave endémica de Nueva Zelanda, la cual se encuentra en peligro de extinción y es famosa por su curiosidad y agresividad. Esta ave da nombre a una extensa colección de algoritmos de Máquinas de conocimiento desarrollados por la universidad de Waikato (Nueva Zelanda) («Sitio oficial de la Universidad de Waikato en Nueva Zelanda», 2019) implementados en Java («Sitio oficial de Java», 2019); útiles para ser aplicados sobre datos mediante los interfaces que ofrece o para embeberlos dentro de cualquier aplicación.

Además Weka («2019», s. f.-c) contiene las herramientas necesarias para realizar transformaciones sobre los datos, tareas de clasificación, regresión, clustering<sup>3</sup>,

---

<sup>3</sup> También llamada agrupamiento, permite la identificación de tipologías o grupos donde los elementos guardan gran similitud entre sí y muchas diferencias con los de otros grupos.

asociación y visualización. Weka está diseñado como una herramienta orientada a la extensibilidad por lo que añadir nuevas funcionalidades es una tarea sencilla. La licencia de Weka es GPL («Sitio oficial de GPL», 2019), lo que significa que este programa es de libre distribución y difusión. Además, ya que Weka está programado en Java, es independiente de la arquitectura, ya que funciona en cualquier plataforma sobre la que haya una máquina virtual Java disponible.

### **2.1.2 MATLAB**

MATLAB, MATrix LABoratory (Laboratorio de Matrices) es un entorno de computación y desarrollo de aplicaciones totalmente integrado orientado para llevar a cabo proyectos en donde se encuentren implicados elevados cálculos matemáticos y la visualización gráfica de los mismos. Integra análisis numérico, cálculo matricial, proceso de señal y visualización gráfica.

MATLAB dispone también en la actualidad de un amplio abanico de programas de apoyo especializados, denominados Toolboxes, que extienden significativamente el número de funciones incorporadas en el programa principal. Estos Toolboxes cubren en la actualidad prácticamente casi todas las áreas principales en el mundo de la ingeniería y la simulación, destacando entre ellos el 'toolbox' de proceso de imágenes, señal, control robusto, estadística, análisis financiero, matemáticas simbólicas, redes neurales, lógica difusa, identificación de sistemas, simulación de sistemas dinámicos, etc. es un entorno de cálculo técnico, que se ha convertido en estándar de la industria, con capacidades no superadas en computación y visualización numérica («Matlab», 2019). Estas características que se han comentado hacen de MATLAB una herramienta de trabajo muy extendida entre los estudiantes, técnicos e investigadores.

En esta investigación se hace uso del MATLAB en su versión r2018a («Sitio oficial de descarga de MATLAB en su versión r2018a», 2019) como herramienta durante las fases: desarrollo del experimento y análisis de los resultados.

## 2.2 Comportamiento de las variables presentes en el problema de investigación

A continuación se expondrán las variables utilizadas durante el proceso de experimentación, las cuales, como se argumentó con anterioridad son las que mayor incidencia tienen sobre el consumo de energía eléctrica de una edificación hotelera. Los valores de las variables ocupacionales (*HDO*<sup>4</sup> y *TDT*<sup>5</sup>) y *CONSUMO TOTAL* son datos reales tomados durante todo el año 2011 en una instalación hotelera de la capital provincial cienfueguera. Los valores de las variables meteorológicas (las demás) son también reales pero tomadas en el Centro de Meteorología de la provincia de Cienfuegos.

**Tabla 1:** Variables presentes en el proceso de experimentación con sus respectivas características

<b>Nomenclador</b>	<b>Nombre completo</b>	<b>Rango de valores</b>	<b>Unidad de medida</b>	<b>Pequeña descripción</b>
<b>Mes</b>	Mes	1 - 12	-	-
<b>Días</b>	Día	1 - 30 ó 1 - 31 según sea la cantidad de días que contenga el mes al que el día en cuestión pertenece, exceptuando a febrero que, en este caso por ser el año 2011 no bisiesto cuenta el mes con un rango de valores de 1 - 28	-	-

<sup>4</sup> Ver HDO en la Tabla 1

<sup>5</sup> Ver TDT en la Tabla 1

<b>HDO</b>	Habitaciones Día Ocupadas	3 - 49	cantidad	Cantidad de habitaciones del hotel ocupadas
<b>TDT</b>	Turistas Día Totales	4 - 99	cantidad	Cantidad total de turistas que se encuentran conviviendo en el hotel
<b>Temp Ext Promedio</b>	Temperatura exterior promedio	18.51 - 29.17	°C <sup>6</sup>	
<b>Temp Máx Promedio</b>	Temperatura máxima promedio	18.61 – 29.29	°C	
<b>Temp Mín Promedio</b>	Temperatura mínima promedio	18.43 - 29.06	°C	
<b>Hum Ext Promedio</b>	Humedad exterior promedio	56.06 – 90.65	% (Por ciento)	
<b>Punto Rocío Promedio</b>	Punto rocío promedio	10.22 - 24.14	°C	
<b>Veloc. Viento Promedio</b>	Velocidad del viento promedio	2.54 – 18.32	m/s <sup>7</sup>	
<b>Presión Promedio</b>	Presión promedio	1003.21 - 1017.69	KPa <sup>8</sup>	
<b>Lluvia Suma</b>	Lluvia suma	0 – 80.51	mm <sup>9</sup>	

<sup>6</sup> Ver °C en el Glosario de Términos

<sup>7</sup> Ver m/s en el Glosario de Términos

<sup>8</sup> Ver KPa en el Glosario de Términos

<sup>9</sup> Ver mm en el Glosario de Términos

<b>Rad. Solar Acumulado</b>	Radiación solar acumulada	4577400 - 27429600	W/m <sup>2</sup> <sup>10</sup>	
<b>CONSUMO TOTAL</b>	Consumo total	956 - 2817	Kw <sup>11</sup>	Consumo total de todas las instalaciones del hotel

## 2.3 Preparación de los datos

El formato de los datos contenidos en la fuente nunca es el idóneo, y la mayoría de las veces no es posible ni siquiera utilizar ningún algoritmo de minería sobre los datos "en bruto". Es por esto que a continuación se procederá a realizar la preparación de los datos, realizada siguiendo el segundo paso de la metodología de minería de datos seleccionada para el desarrollo de esta investigación, KDD.

La granulación temporal del muestreo de las variables climatológicas es de cada 10 minutos del día, obtenidos a partir de históricos anuales del Centro Meteorológico Provincial de Cienfuegos, brindando de esta manera un total de 52 560 datos climatológicos. Mas para su mejor estudio han sido promediadas las variables de cada día para obtener entonces 362 datos climatológicos por variable, correspondiendo con cada día del año 2011, exceptuando el día primero de enero y el 23 y 24 de marzo. Esto pasa pues, en el caso del día primero de enero, para poder contarlo se necesitarían observaciones del día 31 de diciembre del año anterior desde las 9:00 pm pues es cuando se empieza la medición cada 10 minutos; las cuales no existen. En el caso de los días 23 y 24 de marzo tampoco se cuenta con los datos pues no aparecían en el informe brindado por el Centro Meteorológico Provincial de Cienfuegos.

Los datos de las variables climatológicas fueron promediados por cada día para así poder tener datos más acordes a las necesidades de la presente investigación, la cual se centra

<sup>10</sup> Ver W/m<sup>2</sup> en el Glosario de Términos

<sup>11</sup> Ver Kw en el Glosario de Términos

en la predicción del consumo energético en una edificación hotelera en un día futuro dado.

De manera general los datos fueron divididos estratégicamente en dos: Temporada Alta (invierno) y Temporada Baja (verano). Se agruparon en Temporada Baja los meses de abril a octubre. Los demás se agruparon en Temporada Alta. Esto se hizo así pues al promediar por mes los datos de las variables *Habitaciones Día Ocupadas* y *Turistas Día Totales* se notaba una fluctuación apreciable entre los meses de cada temporada. De esta manera para obtener una mayor exactitud en los resultados fueron divididos los datos en dos secciones.

La calidad del conocimiento descubierto no sólo depende del algoritmo de minería utilizado, sino también de la calidad de los datos minados. Por ello, después de la recopilación, el siguiente paso en el proceso de KDD es seleccionar y preparar el subconjunto de datos que se va a minar, los cuales constituyen lo que se conoce como *vista minable*. (Hernández Orallo et al., 2011) De esta manera, siguiendo el tercer paso de la metodología de minería de datos seleccionada: KDD, y haciendo uso de la herramienta Weka se hicieron una serie de tratamientos a los datos, los cuales fueron:

- **Búsqueda de valores ausentes.** La presencia de datos faltantes o perdidos (missing values) puede ser también un problema pernicioso que puede conducir a resultados poco precisos. No obstante, es necesario reflexionar primero sobre el significado de los valores faltantes antes de tomar ninguna decisión sobre cómo tratados ya que éstos pueden deberse a causas muy diversas, como a un mal funcionamiento del dispositivo que hizo la lectura del valor, a cambios efectuados en los procedimientos usados durante la colección de los datos o al hecho de que los datos se recopilen desde fuentes diversas. (Hernández Orallo et al., 2011)  
Como resultado se obtuvo que existían en el conjunto de datos un total de 133 valores ausentes, todos pertenecientes a la variable *Lluvia suma*. Para dar solución a tal problemática, siguiendo la lógica que presenta el conjunto de datos, se decidió insertar por cada uno de estos valores ausentes el valor 0 (cero).
- **Búsqueda de valores erróneos.** Estos datos anómalos pueden representar errores en los datos o pueden ser valores correctos que son simplemente

diferentes a los demás. Algunos algoritmos de minería de datos ignoran estos datos, otros los descartan considerándolos ruido o excepciones, pero otros son muy sensibles y el resultado se ve claramente perjudicado por ello. (Hernández Orallo et al., 2011) Como resultado se obtuvo que no existían en el conjunto de datos valores erróneos.

- **Selección de atributos.** La selección de atributos relevantes es uno de los preprocesamientos más importantes, ya que es crucial que los atributos utilizados sean relevantes para la tarea de minería de datos. Para ello se utilizaron dos de los evaluadores de Weka: CfsSubsetEval («CfsSubsetEval», s. f.) y ReliefFAttributeEval («ReliefFAttributeEval», 2018). De esta manera resultaron 3 datasets por cada temporada, los cuales se muestran en la tabla 2:

**Tabla 2:** Datasets resultantes después de haberse hecho la selección de atributos

Número de dataset	Temporada del dataset	Filtro de selección de atributos	Breve descripción
1	Temporada alta	Ninguno	Dataset de temporada alta con todos sus atributos
2		CfsSubsetEval	Dataset de temporada alta solo con los siguientes atributos: <ul style="list-style-type: none"> <li>• Mes</li> <li>• Temperatura exterior promedio</li> <li>• Temperatura máxima promedio</li> <li>• Temperatura mínima promedio</li> <li>• Humedad exterior promedio</li> <li>• Punto rocío promedio</li> <li>• Veloc. Viento Promedio</li> <li>• Presión Promedio</li> <li>• Radiación solar acumulada</li> </ul>

3		ReliefFAtributeEval	Dataset de temporada alta con todos sus atributos
4	Temporada baja	Ninguno	Dataset de temporada baja con todos sus atributos
5		CfsSubsetEval	Dataset de temporada baja solo con los siguientes atributos: <ul style="list-style-type: none"> <li>• Mes</li> <li>• Turistas Día Totales</li> <li>• Temperatura exterior promedio</li> <li>• Temperatura máxima promedio</li> <li>• Temperatura mínima promedio</li> <li>• Punto rocío promedio</li> </ul>
6		ReliefFAtributeEval	Dataset de temporada baja con todos sus atributos
7	Año completo	Ninguno	Dataset del año completo con todos sus atributos
8		CfsSubsetEval	Dataset del año completo solo con los siguientes atributos: <ul style="list-style-type: none"> <li>• Temperatura exterior promedio</li> <li>• Temperatura máxima promedio</li> <li>• Temperatura mínima promedio</li> <li>• Humedad exterior promedio</li> <li>• Punto rocío promedio</li> </ul>

9		ReliefFAtributeEval	Dataset del año completo con todos sus atributos
---	--	---------------------	--

En muchas técnicas de regresión conviene normalizar las entradas antes de ajustar el modelo ya que su entrenamiento puede dificultarse cuando las entradas tienen escalas muy diferentes. Para ello se utilizó la ecuación:

$$\hat{v} = \frac{v - \text{mín}_A}{\text{máx}_A - \text{mín}_A}$$

donde  $\text{máx}_A$  y  $\text{mín}_A$  son el valor máximo y mínimo que toma la variable A en el dataset; v es el valor actual a normalizar, y  $\hat{v}$  el valor normalizado.

## 2.4 Generalidades sobre la experimentación en MATLAB con los modelos de minería de datos

La experimentación consiste en probar con diferentes modelos de minería de datos con el objetivo de evaluar su efectividad en la predicción del consumo energético en edificaciones hoteleras.

De acuerdo al sexto paso de la metodología KDD, se arriba a seleccionar los métodos de minería de datos a utilizar en la experimentación. Los modelos de ANN a utilizar son MLP, FFNN, LRN, GRNN y NARX. También se experimentó con los DT y los SVM.

De esta manera, siguiendo el séptimo paso de la metodología de minería de datos KDD, se hacen variaciones a los modelos buscando una mayor precisión y adaptabilidad. Para cada modelo se realizan diferentes variaciones como la semilla de generación de números aleatorios para la inicialización de los pesos en el caso de las redes neuronales artificiales (su variación afecta de manera considerable el comportamiento de los modelos de minería de datos), y los datasets para su creación, entrenamiento y prueba. En el caso del modelo MLP se varió también la función de entrenamiento, haciéndose uso de:

entrenamiento de retropropagación BFGS<sup>12</sup> quasi-Newton, entrenamiento de retropropagación con regularización bayesiana, entrenamiento de retropropagación del gradiente conjugado con reinicio Powell-Beale («Conjugate Gradient Algorithms», s. f.), entrenamiento de retropropagación del gradiente conjugado con actualización Fletcher-Reeves («Conjugate Gradient Algorithms», s. f.) y entrenamiento de retropropagación del gradiente descendente.

### **2.4.1 Procedimiento de experimentación**

La experimentación con los modelos de minería de datos queda dividida en cuatro etapas: la primera etapa es para la importación de los datos, en la segunda etapa se crea el modelo, en la tercera etapa se desarrolla el proceso de validación cruzada, y en la cuarta etapa se hace la evaluación del desempeño de los modelos.

#### **Etapa 1: Importación de los datos**

En esta etapa, las variables de entrada y de salida deseada presentan un comportamiento para cada día durante el año 2011, exceptuando el día primero de enero y el 23 y 24 de marzo. Se tienen por lo tanto 362 datos para cada variable. Mas como los datos fueron separados en dos secciones: Temporada Alta y Temporada Baja, se tiene un primer conjunto de 148 datos y un segundo de 214 datos respectivamente.

Estos datos son obtenidos como vectores<sup>13</sup> mediante la función `csvread` ("*alta\_normalizada.csv*"), donde *alta\_normalizada.csv* es el nombre del fichero que contiene los datos a ser procesados. Luego los datos son divididos en dos: la primera parte, compuesta por todo el vector menos la última columna, que contiene el valor de las variables predictoras; y la segunda parte, compuesta por la última columna del vector de los datos, que contiene el valor de la variable objetivo. Por último se configura la semilla de generación de números aleatorios para la inicialización de los pesos, en el caso de las ANN, mediante la función `rng(11)`, donde 11 es el valor de la semilla a utilizar.

#### **Etapa 2: Selección de la técnica**

---

<sup>12</sup> Ver BFGS en el Glosario de Términos

<sup>13</sup> Término usado en MATLAB para referirse a las matrices

Existen varias funciones en MATLAB para la creación de una ANN, un Árbol de Decisión y una Máquina de Soporte Vectorial, según el modelo que se desee. En esta etapa se elige la más apropiada para cada técnica, así como sus argumentos más adecuados.

### **Etapa 3: Desarrollo del proceso de validación cruzada**

Todas las técnicas fueron entrenadas utilizando un procedimiento de validación cruzada. Este de inicia a partir de la sentencia `cvpartition(size(M,1),'KFold',10)`, donde `size(M,1)` devuelve la cantidad de instancias contenidas en el conjunto de datos, `KFold` indica el tipo de validación cruzada a ser utilizada (en este caso, validación cruzada de k iteraciones), y el número `10` señala la cantidad de veces que el conjunto de datos será particionado (y, por lo tanto, iteraciones) y la cantidad de subconjuntos que se crearán con cada partición.

Luego, ya dentro del proceso de validación cruzada, que está conformado en este caso por un iterador de 10 pasos, dentro del cual, como se expresó con anterioridad, se define el DT, la SVM y la GRNN; y a la vez se entrenan todos los modelos.

### **Etapa 4: Evaluación del desempeño de los modelos**

Para evaluar la calidad de los modelos se hace uso del conjunto de prueba donde  $Xi\_test$  está constituido por las variables predictoras y  $yi\_test$  por los valores esperados. Para el caso de las ANN se hizo uso de la sentencia `obtenidos = net(Xi_test)`, donde  $Xi\_test$  es el conjunto de variables predictoras de prueba escogidas en la iteración actual del proceso de validación cruzada, `net` es la red neuronal artificial en cuestión y `obtenidos`, como su nombre lo indica, los resultados obtenidos de la predicción de la ANN. Para el caso del DT y la SVM se hace uso de la función `obtenidos = predict(model,Xi_test)`, donde  $Xi\_test$  es de igual manera el conjunto de variables predictoras de prueba, `obtenidos` los resultados de la predicción, y `model` es el modelo predictivo, sea el árbol de decisión o la máquina de soporte vectorial.

Luego de realizadas las predicciones, se pasa a la medición del desempeño haciendo uso de los resultados de la anterior predicción. De esta manera y haciendo uso de la

```

function [outputArg1] = getRMSE(obt,real)

col=size(obt);

i=col(2);

cv=0;
for n = 1:i;
    cv = cv+(obt(n)-real(n))^2;
end
cv=cv/n;
cv=sqrt(cv);
outputArg1 = cv;

end

```

sentencia  $rmse(i) = getRMSE(obtenidos, y_i\_test)$  se calcula el RMSE<sup>14</sup> del modelo en la iteración actual, donde *obtenidos* es un vector que contiene los datos obtenidos luego de la prueba y, *y<sub>i</sub>\_test* es un vector con los datos que debió de haber ofrecido el modelo de minería de datos durante tal prueba. La función *getRMSE* constituye un algoritmo de creación personal, basado en la ecuación original que calcula el RMSE, que sirve como propuesta de la presente investigación. A continuación, se muestra el código del mencionado algoritmo, creado en el lenguaje utilizado por MATLAB:

Posteriormente, luego de terminadas las 10 iteraciones del proceso de validación cruzada se calcula la media aritmética de los 10 RMSE calculados, mediante la sentencia  $rmsef = mean(rmse)$ .

Luego se finaliza guardando el modelo en la carpeta sobre la cual MATLAB se encuentre trabajando en ese momento (espacio de trabajo de MATLAB). Para ello se utiliza la función de MATLAB `save('pretrained_model_alta.mat', 'tree')`, donde `'pretrained_model_alta.mat'` es el nombre del archivo donde se va a guardar el modelo, y `'tree'` es el nombre de dicho modelo.

---

<sup>14</sup> Ver Criterio de medida del desempeño de los modelos: RMSE.

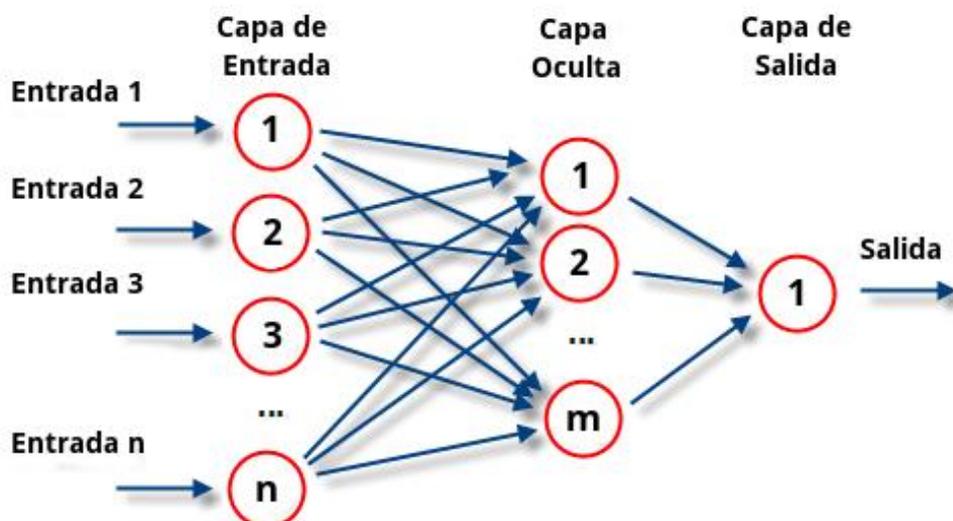
## 2.5 Modelos utilizados en la experimentación

### 2.5.1 Modelo MLP: Teoría y Experimentación

El perceptrón multicapa es una red neuronal artificial formada por múltiples capas, esto le permite resolver problemas que no son linealmente separables, lo cual es la principal limitación del perceptrón (también llamado perceptrón simple). El perceptrón multicapa puede ser totalmente o localmente conectado. En el primer caso cada salida de una neurona de la capa "i" es entrada de todas las neuronas de la capa "i+1", mientras que en el segundo cada neurona de la capa "i" es entrada de una serie de neuronas (región) de la capa "i+1".

La arquitectura de este tipo de red se caracteriza porque tiene todas sus neuronas agrupadas en distintos niveles llamados capas. El primer nivel corresponde a la capa de entrada, que se encarga únicamente de propagar por el resto de la red las entradas recibidas. El último nivel es el de la capa de salida. Se encarga de proporcionar los valores de salida de la red. En las capas intermedias denominadas capas ocultas, se realiza un procesamiento no lineal de los patrones recibidos.

Las conexiones del perceptrón multicapa son hacia adelante. Generalmente todas las neuronas de un nivel se conectan con todas las neuronas de la capa inmediatamente posterior. A veces, dependiendo de la red, se encuentran conexiones de neuronas que no están en niveles consecutivos, o alguna de las conexiones entre dos neuronas de



niveles consecutivos no existe, es decir, el peso asociado a dicha conexión es constante e igual a cero. Además, todas las neuronas de la red tienen un valor umbral asociado. Se suele tratar como una entrada cuyo valor es constante e igual a uno, y lo único que varía es el peso asociado a dicha conexión (que es el umbral realmente). (MORENO RODRÍGUEZ, 2009)

**Fig. 2** Estructura del MLP

Como se ha mencionado antes en cada uno de las variantes del modelo MLP se usaron funciones de entrenamiento diferentes. A continuación, en la Tabla 3 se muestra una relación de cada una de las funciones mencionadas anteriormente con su sintaxis en MATLAB.

**Tabla 3:** Relación de funciones de entrenamiento usadas con su sintaxis en MATLAB

<b>Nombre de la función en MATLAB y su utilidad</b>	<b>Sintaxis en MATLAB</b>
<b><i>trainbr</i> (entrenamiento de retropropagación con regularización bayesiana)</b>	net.trainFcn = ' <i>trainbr</i> ';
<b><i>trainbfg</i> (entrenamiento de retropropagación BFGS quasi-Newton)</b>	net.trainFcn = ' <i>trainbfg</i> ';
<b><i>traincgb</i> (entrenamiento de retropropagación del gradiente conjugado con reinicio Powell-Beale)</b>	net.trainFcn = ' <i>traincgb</i> ';
<b><i>traincgf</i> (entrenamiento de retropropagación del gradiente conjugado con actualización Fletcher-Reeves)</b>	net.trainFcn = ' <i>traincgf</i> ';
<b><i>traingd</i> (entrenamiento de retropropagación del gradiente descendente)</b>	net.trainFcn = ' <i>traingd</i> ';

Para la creación de la ANN de tipo MLP se hizo uso de la función *fitnet(10)*, donde 10 es el número de neuronas que va a contener la capa oculta de la ANN, que en este caso es una sola capa; pero, si se quisiera crear una ANN con más de una capa, se utilizaría como argumento de la función un vector con longitud igual a la cantidad de capas ocultas

deseadas, ubicando en cada posición la cantidad de neuronas que se deseen colocar en cada capa.

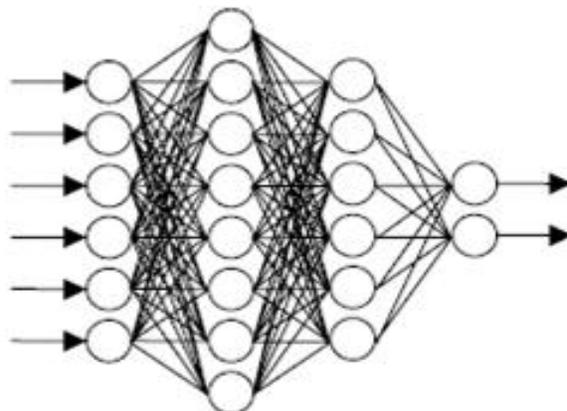
Como resultado de la experimentación en MATLAB con el modelo MLP se obtuvieron 225 variantes del mismo, las cuales se detallan en el Anexo 1.

### 2.5.2 Modelo FFNN: Teoría y Experimentación

Como su nombre lo indica, en este tipo de redes se empieza con un vector de entradas el cual es equivalente en magnitud al número de neuronas de la primera capa de la red, las cuales procesan dicho vector elemento por elemento en paralelo. La información, modificada por los factores multiplicativos de los pesos en cada neurona, es transmitida hacia delante por la red pasando por las capas ocultas (si hay) para finalmente ser procesada por la capa de salida. Es por eso que este tipo de redes reciben su nombre.

Es importante mencionar que las redes feedforward son las más sencillas en cuanto a implementación y simulación, pero su desempeño es bueno para aplicaciones en los que no se requiera que la red retenga información de eventos pasados como ayuda para evaluar eventos futuros. Cada vector de entrada presentado como entrenamiento para este tipo de redes es una entidad aislada del resto y, al final de dicho periodo de prueba, la red estará lista para comenzar a identificar y clasificar patrones, reconocer imágenes o cualquier otra aplicación que se le quiera dar.

Al iniciar las investigaciones sobre redes neuronales, las redes feedforward fueron las que recibieron más atención de parte de los investigadores porque sus características en cuanto a tiempos de procesamiento hacían viables simulaciones con los equipos



computacionales de la época. Comparadas con las otras redes, las FFNN (Feedforward Neural Network) son una opción cuyo balance costo-velocidad y costo-exactitud es tal que da mayor ventaja al costo que a los otros parámetros. En este tipo de redes no existen interconexiones entre capas más allá de la conexión directa hacia adelante para propagar la información. No hay rutas de retroalimentación para desempeñar la función de memoria de la red. (Oropeza, s. f.)

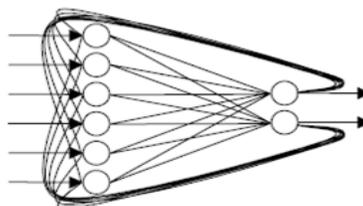
**Fig. 3** Estructura de una ANN feedforward. (Arencibia, 2018)

Para la creación de las ANN de tipo FFNN y FFNN en cascada se hizo uso de las funciones *feedforwardnet(10)* y *cascadeforwardnet(10)* respectivamente; donde 10 es el número de neuronas que va a contener la capa oculta de la ANN, que en ambos casos es una sola capa. Igualmente, si se quisiera crear una ANN con más de una capa, se utilizaría como argumento de la función un vector con longitud igual a la cantidad de capas ocultas deseadas, ubicando en cada posición la cantidad de neuronas que se deseen colocar en cada capa.

Como resultado de la experimentación en MATLAB con el modelo FFNN se obtuvieron 90 variantes del mismo, las cuales se detallan en el Anexo 2.

### 2.5.3 Modelo LRN: Teoría y Experimentación

Las redes neuronales recurrentes completamente conectadas, a diferencia de las redes Elman<sup>15</sup>, tienen caminos de retroalimentación entre todos los elementos que las conforman. Una sola neurona está entonces conectada a las neuronas posteriores en la siguiente capa, las neuronas pasadas de la capa anterior y a ella misma a través de vectores de pesos variables que sufren alteraciones en cada iteración con el fin de alcanzar los parámetros o metas de operación. La complejidad de este tipo de redes es



<sup>15</sup> Las ANN Elman son redes neuronales artificiales feedforward con conexiones recurrentes entre las capas que cuentan con un cierto retraso.

alta en comparación con una red feedforward, por ejemplo, ya que en esta última la red sólo es capaz de transmitir la información hacia las capas siguientes resultando en un efecto de propagación hacia atrás en el tiempo. Las redes neuronales recurrentes, en cambio, realizan el intercambio de información entre neuronas de una manera mucho más compleja y por sus características, dependiendo del tipo de algoritmo de entrenamiento que se elija, pueden propagar la información hacia delante en el tiempo, lo cual equivale a predecir eventos. Esta es una característica muy importante para ciertas aplicaciones, como los IDS<sup>16</sup>, ya que la capacidad de predicción de eventos significativos (en este caso, ataques a la red) basada en las entradas anteriores al sistema le proporciona un beneficio importante a la seguridad del mismo. (Oropeza, s. f.)

**Fig. 4** Estructura de una red recurrente. (Arencibia, 2018)

Para la creación de la ANN de tipo LRN se hizo uso de la función *layreclnet(1:1, 10)*, donde *1:1* es el retraso asociado a la capa oculta y *10* es el número de neuronas que va a contener tal capa oculta de la ANN, que en este caso es una sola capa; pero, si se quisiera crear una ANN con más de una capa, se utilizaría como argumento de la función un vector con longitud igual a la cantidad de capas ocultas deseadas, ubicando en cada posición la cantidad de neuronas que se deseen colocar en cada capa.

Como resultado de la experimentación en MATLAB con el modelo LRN se obtuvieron 45 variantes del mismo, las cuales se detallan en el Anexo 3.

#### **2.5.4 Modelo GRNN: Teoría y Experimentación**

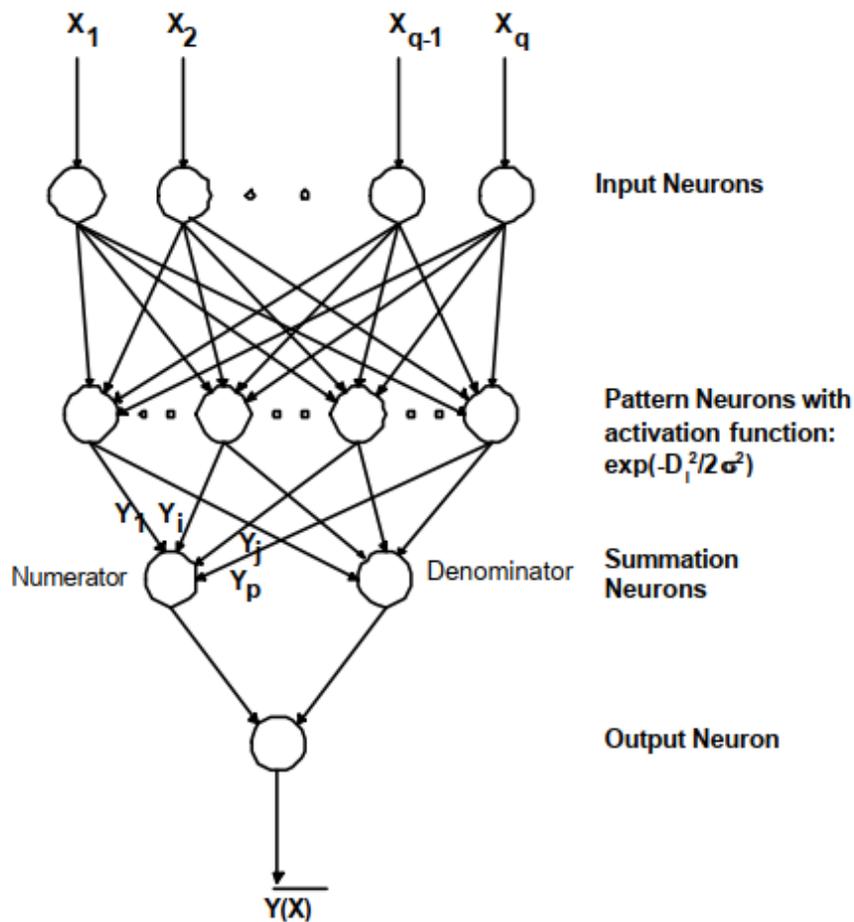
Un modelo GRNN es un tipo de red neuronal con función de base radial (RBF, por sus siglas en inglés). La red neuronal regresiva generalizada clásica es un tipo de red neuronal de base radial. Tiene una fuerte capacidad de mapeo y flexibilidad, y la estructura de la red tiene un alto grado de tolerancia a fallas y robustez. GRNN consiste en una red de cuatro capas, es decir, la capa de entrada, la capa de modo, la capa de suma y la capa de salida. (Chen et al., 2019) La GRNN cae dentro de la categoría de red neuronal artificial probabilística. Esta ANN como otras ANN probabilísticas necesita solo

---

<sup>16</sup> Intrusion Detection System (Sistema de detección de intrusos).

una fracción de las muestras de entrenamiento que una red neuronal artificial de propagación hacia atrás necesitaría. Esto supone una ventaja en casos (como el del presente proyecto) donde la cantidad de instancias disponibles para ser utilizadas en el proceso de experimentación es relativamente pequeña.

Para la creación de la ANN de tipo GRNN se hizo uso de la función  $newgrnn(Xi\_train, yi\_train)$ , donde  $Xi\_train$  es el conjunto de datos de entrada y  $yi\_train$  es el conjunto de datos de salida.

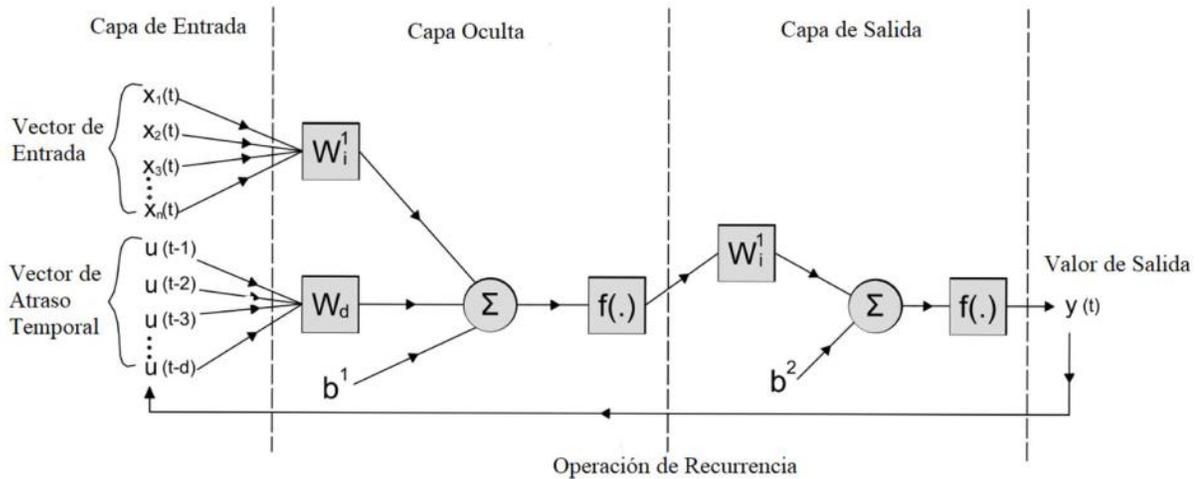


**Fig. 5** Estructura de una GRNN.

Como resultado de la experimentación en MATLAB con el modelo GRNN se obtuvieron 45 variantes del mismo, las cuales se detallan en el Anexo 4.

### 2.5.5 Modelo NARX: Teoría y Experimentación

La red auto regresiva no lineal con aportes exógenos (nonlinear autoregressive network with exogenous inputs - NARX) es una red dinámica parcialmente recurrente, con conexiones de información retroactiva incluyendo varias capas de la red.



**Fig. 6** Esquema general de una ANN NARX con una capa oculta. (A. Sierra, Manzi, M. Brentan, & Luvizotto Jr., 2017)

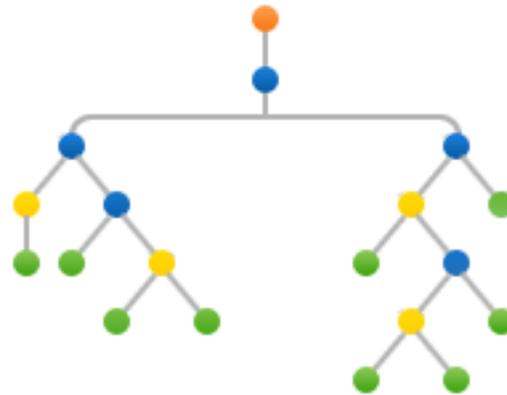
Para la creación de la ANN de tipo NARX se hizo uso de la función *narxnet(1:2,1:2,15)*, donde el primer argumento hace referencia al retraso asociado a las entradas de la capa oculta, el segundo argumento se refiere al retraso de reacción de la capa oculta, y el tercer argumento hace referencia a la cantidad de neuronas que se encuentran dentro de la capa oculta, que en este caso es una sola capa; pero, si se quisiera crear una ANN con más de una capa, se utilizaría como argumento de la función un vector con longitud igual a la cantidad de capas ocultas deseadas, ubicando en cada posición la cantidad de neuronas que se deseen colocar en cada capa.

Como resultado de la experimentación en MATLAB con el modelo NARX se obtuvieron 45 variantes del mismo, las cuales se detallan en el Anexo 5.

## 2.5.6 Modelo Árbol de Decisión: Teoría y Experimentación

Un árbol tiene muchas analogías en la vida real, y resulta que ha influido en una amplia área del aprendizaje automático o Machine Learning. Los árboles de decisión son una técnica de aprendizaje supervisado que predice valores de respuestas mediante el aprendizaje de reglas de decisión derivadas de características.

Los árboles de decisión funcionan al dividir el espacio de la característica en varias regiones rectangulares simples, divididas por divisiones paralelas de ejes. Para obtener una predicción para una observación particular, se utiliza la media o el modo de las respuestas de las observaciones de entrenamiento, dentro de la partición a la que pertenece la nueva observación.



**Fig. 7** Diagrama de un DT

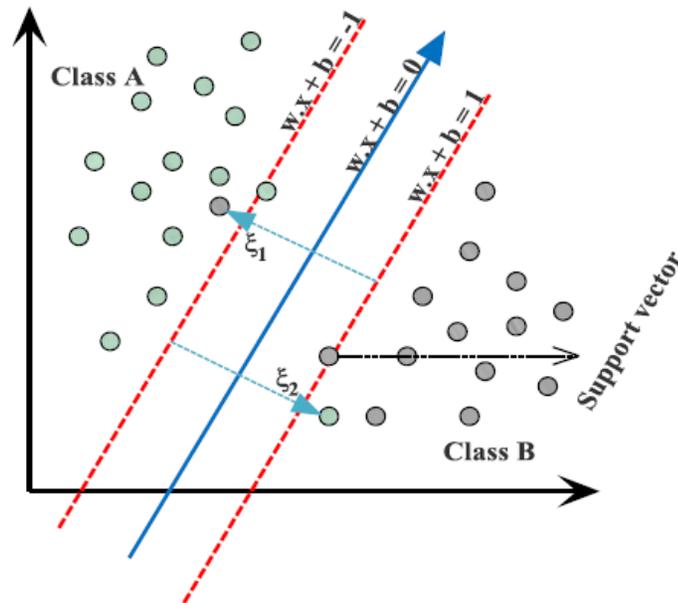
En la creación del DT se hizo uso de la función `fitrtree(Xi_train,yi_train)`, en la cual `Xi_train` es el conjunto de datos de entrada y `yi_train` es el conjunto de datos objetivo.

Como resultado de la experimentación en MATLAB con el modelo DT se obtuvieron 45 variantes del mismo, las cuales se detallan en el Anexo 6.

## 2.5.7 Modelo Máquina de Soporte de Vectorial: Teoría y Experimentación

Una Máquina de Soporte Vectorial aprende la superficie decisión de dos clases distintas de los puntos de entrada. Como un clasificador de una sola clase, la descripción dada por los datos de los vectores de soporte es capaz de formar una frontera de decisión

alrededor del dominio de los datos de aprendizaje con muy poco o ningún conocimiento de los datos fuera de esta frontera. Los datos son mapeados por medio de un kernel



Gaussiano u otro tipo de kernel a un espacio de características en un espacio dimensional más alto, donde se busca la máxima separación entre clases. Esta función de frontera, cuando es traída de regreso al espacio de entrada, puede separar los datos en todas las clases distintas, cada una formando un agrupamiento. (Betancourt, 2010)

**Fig. 8** SVM regresora

Para la creación de la máquina de soporte vectorial se utilizó la función  $\text{fitrsvm}(X_i\_train, y_i\_train)$ , en la cual el primer argumento es el conjunto de datos de entrada y el segundo argumento es el conjunto de datos objetivo.

Como resultado de la experimentación en MATLAB con el modelo SVM se obtuvieron 45 variantes del mismo, las cuales se detallan en el Anexo 7.

## Conclusiones del capítulo

- Las herramientas empleadas para el desarrollo de los experimentos son Weka y MATLAB.
- Los modelos más apropiados para la predicción en este problema son: MLP, FFNN, LRN, GRNN, NARX, Árbol de Decisión y Máquina de Soporte Vectorial.

- Los experimentos se organizaron en 4 etapas: la primera etapa es para la importación de los datos, en la segunda etapa se crea el modelo, en la tercera etapa se desarrolla el proceso de validación cruzada, y en la cuarta etapa se hace la evaluación del desempeño de los modelos.
- Se experimentaron con 225 variantes del MLP, 90 variantes del FFNN, 45 variantes del LRN, 45 variantes del GRNN, 45 variantes del NARX, 45 variantes del DT y 45 variantes del SVM.

## Capítulo III: Análisis de los resultados.

En el presente capítulo, siguiendo el octavo paso de KDD, se realiza un análisis de los resultados obtenidos luego de la experimentación desarrollada en el capítulo anterior. Primeramente, se detalla el criterio de medida de desempeño utilizado, el RMSE; así como los resultados obtenidos de las variantes de cada modelo. Luego se selecciona el mejor resultado de cada dataset de acuerdo al mencionado criterio. Finalmente se define, entre todos los modelos, el mejor para la predicción del consumo de energía eléctrica de una edificación hotelera.

### 3.1 Criterio de medida del desempeño de los modelos: RMSE.

Durante el proceso de experimentación, para cada uno de los modelos seleccionados, es necesario obtener una medida del desempeño de dichos modelos. Para ello se utiliza el criterio RMSE, detallado a continuación.

Se define RMSE (Root Mean Square Error) como la raíz cuadrada de la media de los errores al cuadrado:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2}$$

donde  $n$  es la cantidad de predicciones,  $y_t$  es el valor deseado y  $\hat{y}_t$  el valor obtenido.

El error cuadrático medio (RMSE) también llamado desviación cuadrática media es una medida de uso frecuente de la diferencia entre los valores pronosticados por un modelo y los valores realmente observados. Estas diferencias individuales son también llamadas residuos y el RMSE sirve para agregar en una sola medida la capacidad de predicción. (Vélez Correa & Nieto Figueroa, 2016) Es presentado por Rob J. Hyndman («Rob J Hyndman», 2019) y Anne Koehler («Anne Koehler», s. f.) en (J. Hyndman & B. Koehler, s. f.) y por Christoph Bergmeir («Christoph Bergmeir», s. f.) y José M. Benítez («José Manuel Benítez Sánchez», 2019) en (Bergmeir & Benítez, 2012). Es igualmente usado en (Liu, Li, & Q. Tian, 2010), (Karamirad, Omid, Alimardani, Mousazadeh, & Heidari, 2013) y (Gaiser, T., de Barros, I., Sereke, F., & Lange, F. M., 2010).

## 3.2 Selección de la mejor variante de los modelos implementados por cada dataset según el criterio RMSE

Luego de realizar las etapas de la experimentación en cada uno de las variantes de los modelos seleccionados, y luego de aplicar el criterio de medida de desempeño en cada variante, se obtiene los resultados finales de cada una de estas.

### 3.2.1 Con el dataset de la temporada alta

#### 3.2.1.1 Haciendo uso del dataset completo

##### 3.2.1.1.1 Modelo MLP

La tabla 4 muestra la variante del modelo MLP, entrenado con el dataset de temporada alta sin selección de atributos, que mejores resultados obtuvo; de un total de 25 variantes desarrolladas, las cuales se presentan en el anexo 8.

**Tabla 4:** Variante con mejores resultados del modelo MLP entrenado con el dataset de temporada alta sin selección de atributos.

Variante	Algoritmo de entrenamiento	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
2	entrenamiento de retropropagación BFGS quasi-Newton	3	0.093066	0.093539

##### 3.2.1.1.2 Modelo FFNN

La tabla 5 muestra las variantes del modelo FFNN, entrenado con el dataset de temporada alta sin selección de atributos, que mejores resultados obtuvo; de un total de 10 variantes desarrolladas, las cuales se presentan en el anexo 9. Nótese que la Variante 8 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 3 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 5:** Variantes con mejores resultados del modelo FFNN entrenado con el dataset de temporada alta sin selección de atributos.

Variante	Arquitectura	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>3</b>	FFNN	5	0.083821	0.082301
<b>8</b>	FFNN en cascada	5	0.079252	0.09182

### 3.2.1.1.3 Modelo LRN

La tabla 6 muestra los resultados obtenidos de cada una de las variantes del modelo LRN, entrenado con el dataset de temporada alta. Nótese que la Variante 4 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 2 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 6:** Resultados de las variantes del modelo LRN entrenado con el dataset de temporada alta sin selección de atributos.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>1</b>	1	0.00010315	0.25466
<b>2</b>	<b>3</b>	<b>0.025942</b>	<b>0.22644</b>
<b>3</b>	5	0.0029316	0.23956
<b>4</b>	<b>11</b>	<b>2.0524e-09</b>	<b>0.28658</b>
<b>5</b>	17	0.0047173	0.2536

### 3.2.1.1.4 Modelo GRNN

La tabla 7 muestra los resultados obtenidos de cada una de las variantes del modelo GRNN, entrenado con el dataset de temporada alta. Nótese que la Variante 4 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 2 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 7:** Resultados de las variantes del modelo GRNN entrenado con el dataset de temporada alta sin selección de atributos.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
1	1	6.78e-08	0.11685
2	3	7.4726e-08	0.10567
3	5	2.0339e-07	0.11138
4	11	5.9983e-08	0.11775
5	17	2.3449e-07	0.11367

### 3.2.1.1.5 Modelo NARX

La tabla 8 muestra los resultados obtenidos de cada una de las variantes del modelo NARX, entrenado con el dataset de temporada alta. Nótese que la Variante 5 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 3 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 8:** Resultados de las variantes del modelo NARX entrenado con el dataset de temporada alta sin selección de atributos.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
1	1	0.08965	0.163
2	3	0.089748	0.15776
3	5	0.10179	0.15308
4	11	0.096089	0.16563
5	17	0.086892	0.15823

### 3.2.1.1.6 Modelo DT

La tabla 9 muestra los resultados obtenidos de cada una de las variantes del modelo DT, entrenado con el dataset de temporada alta. Nótese que la Variante 1 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en

la etapa de entrenamiento; mas es la Variante 2 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 9:** Resultados de las variantes del modelo DT entrenado con el dataset de temporada alta sin selección de atributos.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>1</b>	<b>1</b>	<b>0.012079</b>	<b>0.089708</b>
<b>2</b>	<b>3</b>	<b>0.018633</b>	<b>0.060149</b>
<b>3</b>	5	0.028617	0.066189
<b>4</b>	11	0.019935	0.078618
<b>5</b>	17	0.025537	0.067536

### 3.2.1.1.7 Modelo SVM

La tabla 10 muestra los resultados obtenidos de cada una de las variantes del modelo SVM, entrenado con el dataset de temporada alta. Nótese que la Variante 3 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 5 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 10:** Resultados de las variantes del modelo SVM entrenado con el dataset de temporada alta sin selección de atributos.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>1</b>	1	0.10276	0.10741
<b>2</b>	3	0.10316	0.084077
<b>3</b>	<b>5</b>	<b>0.10088</b>	<b>0.083497</b>
<b>4</b>	11	0.10583	0.10505
<b>5</b>	<b>17</b>	<b>0.10464</b>	<b>0.075437</b>

### 3.2.1.2 Haciendo uso del dataset luego de haber hecho la selección de atributos mediante el evaluador de Weka CfsSubsetEval

#### 3.2.1.2.1 Modelo MLP

La tabla 11 muestra las variantes del modelo FFNN, entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka CfsSubsetEval, que mejores resultados obtuvo; de un total de 10 variantes desarrolladas, las cuales se presentan en el anexo 10. Nótese que la Variante 35 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 31 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 21:** Variantes con mejores resultados del modelo MLP entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Algoritmo de entrenamiento	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
31	entrenamiento de retropropagación con regularización bayesiana	1	0.096354	0.092448
35		17	0.09624	0.098735

#### 3.2.1.2.2 Modelo FFNN

La tabla 12 muestra la variante del modelo FFNN, entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka CfsSubsetEval, que mejores resultados obtuvo; de un total de 10 variantes desarrolladas, las cuales se presentan en el anexo 11.

**Tabla 32:** Variante con mejores resultados del modelo FFNN entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Arquitectura	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
14	FFNN	11	0.091184	0.0933

### 3.2.1.2.3 Modelo LRN

La tabla 13 muestra los resultados obtenidos de cada una de las variantes del modelo LRN, entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka CfsSubsetEval. Nótese que la Variante 6 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 7 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 43:** Resultados de las variantes del modelo LRN entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>6</b>	<b>1</b>	<b>0.033157</b>	<b>0.18177</b>
<b>7</b>	<b>3</b>	<b>0.05201</b>	<b>0.15426</b>
<b>8</b>	5	0.054043	0.17727
<b>9</b>	11	0.046	0.25036
<b>10</b>	17	0.034046	0.18392

### 3.2.1.2.4 Modelo GRNN

La tabla 14 muestra los resultados obtenidos de cada una de las variantes del modelo GRNN, entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka CfsSubsetEval. Nótese que la Variante 7 señalada en azul, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset.

**Tabla 54:** Resultados de las variantes del modelo GRNN entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>6</b>	1	9.4649e-09	0.17939
<b>7</b>	<b>3</b>	<b>4.3814e-09</b>	<b>0.17929</b>

<b>8</b>	5	7.9218e-09	0.20726
<b>9</b>	11	1.2614e-08	0.19802
<b>10</b>	17	4.5982e-08	0.18316

### 3.2.1.2.5 Modelo NARX

La tabla 15 muestra los resultados obtenidos de cada una de las variantes del modelo NARX, entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka CfsSubsetEval. Nótese que la Variante 7 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 10 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 65:** Resultados de las variantes del modelo NARX entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

<b>Variante</b>	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>6</b>	1	0.094526	0.1615
<b>7</b>	<b>3</b>	<b>0.092383</b>	<b>0.17209</b>
<b>8</b>	5	0.1033	0.16943
<b>9</b>	11	0.099125	0.18183
<b>10</b>	<b>17</b>	<b>0.094923</b>	<b>0.15851</b>

### 3.2.1.2.6 Modelo DT

La tabla 16 muestra los resultados obtenidos de cada una de las variantes del modelo DT, entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka CfsSubsetEval. Nótese que la Variante 9 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 8 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 76:** Resultados de las variantes del modelo DT entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
6	1	0.024823	0.11214
7	3	0.045514	0.1243
8	5	0.041286	0.086782
9	11	0.011189	0.11124
10	17	0.036466	0.12755

### 3.2.1.2.7 Modelo SVM

La tabla 17 muestra los resultados obtenidos de cada una de las variantes del modelo SVM, entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka CfsSubsetEval. Nótese que la Variante 7 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 10 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 87:** Resultados de las variantes del modelo SVM entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
6	1	0.1255	0.11016
7	3	0.12406	0.091753
8	5	0.12587	0.099152
9	11	0.12614	0.10286
10	17	0.12722	0.087256

### 3.2.1.3 Haciendo uso del dataset luego de haber hecho la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval

#### 3.2.1.3.1 Modelo MLP

La tabla 18 muestra la variante del modelo MLP, entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval, que mejores resultados obtuvo; de un total de 25 variantes desarrolladas, las cuales se presentan en el anexo 12.

**Tabla 98:** Variante con mejores resultados del modelo MLP entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval.

Variante	Algoritmo de entrenamiento	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>57</b>	entrenamiento de retropropagación con regularización bayesiana	3	0.093923	0.093031

#### 3.2.1.3.2 Modelo FFNN

La tabla 19 muestra la variante del modelo FFNN, entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval, que mejores resultados obtuvo; de un total de 10 variantes desarrolladas, las cuales se presentan en el anexo 13.

**Tabla 109:** Variante con mejores resultados del modelo FFNN entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval.

Variante	Arquitectura	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>26</b>	FFNN en cascada	1	0.085151	0.080983

### 3.2.1.3.3 Modelo LRN

La tabla 20 muestra los resultados obtenidos de cada una de las variantes del modelo LRN, entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval. Nótese que la Variante 15 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 13 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 20:** Resultados de las variantes del modelo LRN entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
11	1	0.013916	0.22287
12	3	0.008687	0.22584
13	5	0.0017865	0.206
14	11	1.6954e-09	0.23507
15	17	9.0511e-11	0.31328

### 3.2.1.3.4 Modelo GRNN

La tabla 21 muestra los resultados obtenidos de cada una de las variantes del modelo GRNN, entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval. Nótese que la Variante 14 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 12 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 211:** Resultados de las variantes del modelo GRNN entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
----------	---------------------------------	-----------------------------------	----------------------------

<b>11</b>	1	6.78e-08	0.11685
<b>12</b>	<b>3</b>	<b>7.4726e-08</b>	<b>0.10567</b>
<b>13</b>	5	2.0339e-07	0.11138
<b>14</b>	<b>11</b>	<b>5.9984e-08</b>	<b>0.11775</b>
<b>15</b>	17	2.3449e-07	0.11367

### 3.2.1.3.5 Modelo NARX

La tabla 22 muestra los resultados obtenidos de cada una de las variantes del modelo NARX, entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval. Nótese que la Variante 11 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 15 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 22:** Resultados de las variantes del modelo NARX entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>11</b>	<b>1</b>	<b>0.088195</b>	<b>0.16213</b>
<b>12</b>	3	0.095507	0.17737
<b>13</b>	5	0.098395	0.17668
<b>14</b>	11	0.099791	0.15367
<b>15</b>	<b>17</b>	<b>0.088877</b>	<b>0.13732</b>

### 3.2.1.3.6 Modelo DT

La tabla 23 muestra los resultados obtenidos de cada una de las variantes del modelo DT, entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval. Nótese que la Variante 11 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 12 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 23:** Resultados de las variantes del modelo DT entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka ReliefFAttributeEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>11</b>	<b>1</b>	<b>0.012079</b>	<b>0.089708</b>
<b>12</b>	<b>3</b>	<b>0.018634</b>	<b>0.060149</b>
<b>13</b>	5	0.028618	0.066189
<b>14</b>	11	0.019936	0.078619
<b>15</b>	17	0.025537	0.067536

### 3.2.1.3.7 Modelo SVM

La tabla 24 muestra los resultados obtenidos de cada una de las variantes del modelo SVM, entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka ReliefFAttributeEval. Nótese que la Variante 13 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 15 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 24:** Resultados de las variantes del modelo SVM entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka ReliefFAttributeEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>11</b>	1	0.10298	0.10745
<b>12</b>	3	0.1032	0.084329
<b>13</b>	<b>5</b>	<b>0.10167</b>	<b>0.083263</b>
<b>14</b>	11	0.10586	0.10461
<b>15</b>	<b>17</b>	<b>0.10414</b>	<b>0.074922</b>

### 3.2.1.4 Conclusiones del epígrafe 3.2.1

Luego de aplicado el criterio RMSE a cada una de las variantes de los modelos, entrenados con los 3 datasets de temporada alta usados, se le obtiene la medida del desempeño de cada uno de estas variantes y se transita a seleccionar a los que mejor criterio RMSE tienen, los cuales se presentan en la tabla 25.

**Tabla 25:** Mejor resultado por cada dataset de la temporada alta.

Filtro de selección de atributos	Modelo	Variante	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>Ninguno</b>	DT	2	0.018633	0.060149
<b>CfsSubsetEval</b>	DT	8	0.041286	0.086782
<b>ReliefFAtributeEval</b>	DT	12	0.018634	0.060149

## 3.2.2 Con el dataset de la temporada baja

### 3.2.2.1 Haciendo uso del dataset completo

#### 3.2.2.1.1 Modelo MLP

La tabla 26 muestra las variantes del modelo MLP, entrenado con el dataset de temporada baja sin selección de atributos, que mejores resultados obtuvo; de un total de 25 variantes desarrolladas, las cuales se presentan en el anexo 14. Nótese que la Variante 77 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 76 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 26:** Variantes con mejores resultados del modelo MLP entrenado con el dataset de temporada baja sin selección de atributos.

Variante	Algoritmo de entrenamiento	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>76</b>	entrenamiento de retropropagación BFGS quasi-Newton	1	0.10729	0.10452
<b>77</b>		3	0.10636	0.10636

### 3.2.2.1.2 Modelo FFNN

La tabla 27 muestra la variante del modelo FFNN, entrenado con el dataset de temporada baja sin selección de atributos, que mejores resultados obtuvo; de un total de 10 variantes desarrolladas, las cuales se presentan en el anexo 15.

**Tabla 27:** Variante con mejores resultados del modelo FFNN entrenado con el dataset de temporada baja sin selección de atributos.

Variante	Arquitectura	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>40</b>	FFNN en cascada	17	0.074651	0.07626

### 3.2.2.1.3 Modelo LRN

La tabla 28 muestra los resultados obtenidos de cada una de las variantes del modelo LRN, entrenado con el dataset de temporada baja. Nótese que la Variante 17 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 16 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 28:** Resultados de las variantes del modelo LRN entrenado con el dataset de temporada baja sin selección de atributos.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>16</b>	<b>1</b>	<b>0.027786</b>	<b>0.1748</b>
<b>17</b>	<b>3</b>	<b>0.023451</b>	<b>0.23155</b>
<b>18</b>	5	0.029056	0.26732
<b>19</b>	11	0.030382	0.17608
<b>20</b>	17	0.035377	0.18012

### 3.2.2.1.4 Modelo GRNN

La tabla 29 muestra los resultados obtenidos de cada una de las variantes del modelo GRNN, entrenado con el dataset de temporada baja. Nótese que la Variante 16 señalada

en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 19 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 29:** Resultados de las variantes del modelo GRNN entrenado con el dataset de temporada baja sin selección de atributos.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
16	1	5.2469e-08	0.10261
17	3	1.2859e-07	0.098157
18	5	9.8222e-08	0.096273
19	11	1.3623e-07	0.096225
20	17	1.4626e-07	0.099163

### 3.2.2.1.5 Modelo NARX

La tabla 30 muestra los resultados obtenidos de cada una de las variantes del modelo NARX, entrenado con el dataset de temporada baja. Nótese que la Variante 11 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 20 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 30:** Resultados de las variantes del modelo NARX entrenado con el dataset de temporada baja sin selección de atributos.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
16	1	0.092689	0.15025
17	3	0.083674	0.14292
18	5	0.087377	0.15202
19	11	0.076887	0.15521
20	17	0.092659	0.13836

### 3.2.2.1.6 Modelo DT

La tabla 31 muestra los resultados obtenidos de cada una de las variantes del modelo DT, entrenado con el dataset de temporada baja. Nótese que la Variante 20 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 16 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 31:** Resultados de las variantes del modelo DT entrenado con el dataset de temporada baja sin selección de atributos.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>16</b>	<b>1</b>	<b>0.031426</b>	<b>0.033936</b>
<b>17</b>	3	0.026914	0.070328
<b>18</b>	5	0.024492	0.08482
<b>19</b>	11	0.02484	0.038952
<b>20</b>	<b>17</b>	<b>0.022142</b>	<b>0.046679</b>

### 3.2.2.1.7 Modelo SVM

La tabla 32 muestra los resultados obtenidos de cada una de las variantes del modelo SVM, entrenado con el dataset de temporada baja. Nótese que la Variante 17 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 20 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 32:** Resultados de las variantes del modelo SVM entrenado con el dataset de temporada baja sin selección de atributos.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>16</b>	1	0.021506	0.050522
<b>17</b>	<b>3</b>	<b>0.01761</b>	<b>0.064682</b>
<b>18</b>	5	0.018527	0.086272
<b>19</b>	11	0.018186	0.052055
<b>20</b>	<b>17</b>	<b>0.01915</b>	<b>0.041187</b>

### 3.2.2.2 Haciendo uso del dataset luego de haber hecho la selección de atributos mediante el evaluador de Weka CfsSubsetEval

#### 3.2.2.2.1 Modelo MLP

La tabla 33 muestra las variantes del modelo MLP, entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka CfsSubsetEval, que mejores resultados obtuvo; de un total de 25 variantes desarrolladas, las cuales se presentan en el anexo 16. Nótese que la Variante 106 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 108 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 33:** Variantes con mejores resultados del modelo MLP entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Algoritmo de entrenamiento	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
106	entrenamiento de retropropagación con regularización bayesiana	1	0.070716	0.10406
108		5	0.072287	0.1009

#### 3.2.2.2.2 Modelo FFNN

La tabla 34 muestra la variante del modelo FFNN, entrenado con el dataset de temporada baja sin selección de atributos, que mejores resultados obtuvo; de un total de 10 variantes desarrolladas, las cuales se presentan en el anexo 17.

**Tabla 34:** Variante con mejores resultados del modelo FFNN entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Arquitectura	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
----------	--------------	---------------------------------	-----------------------------------	----------------------------

<b>43</b>	FFNN	5	0.077559	0.078644
-----------	------	---	----------	----------

### 3.2.2.2.3 Modelo LRN

La tabla 35 muestra los resultados obtenidos de cada una de las variantes del modelo LRN, entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka CfsSubsetEval. Nótese que la Variante 23 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 25 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 35:** Resultados de las variantes del modelo LRN entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>21</b>	1	0.04842	0.17753
<b>22</b>	3	0.052081	0.17397
<b>23</b>	<b>5</b>	<b>0.040432</b>	<b>0.21325</b>
<b>24</b>	11	0.047221	0.18001
<b>25</b>	<b>17</b>	<b>0.054099</b>	<b>0.17112</b>

### 3.2.2.2.4 Modelo GRNN

La tabla 36 muestra los resultados obtenidos de cada una de las variantes del modelo GRNN, entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka CfsSubsetEval. Nótese que la Variante 23 señalada en azul, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset.

**Tabla 36:** Resultados de las variantes del modelo GRNN entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
21	1	9.2054e-10	0.34001
22	3	0.0076985	0.31466
23	5	3.801e-10	0.30443
24	11	3.161e-09	0.30917
25	17	9.1487e-09	0.32445

### 3.2.2.2.5 Modelo NARX

La tabla 37 muestra los resultados obtenidos de cada una de las variantes del modelo NARX, entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka CfsSubsetEval. Nótese que la Variante 24 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 25 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 37:** Resultados de las variantes del modelo NARX entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
21	1	0.082075	0.13896
22	3	0.093045	0.13034
23	5	0.08199	0.14556
24	11	0.079507	0.13676
25	17	0.083767	0.12952

### 3.2.2.2.6 Modelo DT

La tabla 38 muestra los resultados obtenidos de cada una de las variantes del modelo DT, entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka CfsSubsetEval. Nótese que la Variante 21 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 24 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 38:** Resultados de las variantes del modelo DT entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>21</b>	<b>1</b>	<b>0.026263</b>	<b>0.062617</b>
<b>22</b>	3	0.033311	0.072797
<b>23</b>	5	0.028868	0.077826
<b>24</b>	<b>11</b>	<b>0.026915</b>	<b>0.056615</b>
<b>25</b>	17	0.03944	0.061689

### 3.2.2.2.7 Modelo SVM

La tabla 39 muestra los resultados obtenidos de cada una de las variantes del modelo SVM, entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka CfsSubsetEval. Nótese que la Variante 21 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 25 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 39:** Resultados de las variantes del modelo SVM entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
----------	---------------------------------	-----------------------------------	----------------------------

<b>21</b>	<b>1</b>	<b>0.0037503</b>	<b>0.051196</b>
<b>22</b>	3	0.0062864	0.071375
<b>23</b>	5	0.0045483	0.090349
<b>24</b>	11	0.0073516	0.059813
<b>25</b>	<b>17</b>	<b>0.0094611</b>	<b>0.044869</b>

### 3.2.2.3 Haciendo uso del dataset luego de haber hecho la selección de atributos mediante el evaluador de Weka ReliefFAttributeEval

#### 3.2.2.3.1 Modelo MLP

La tabla 40 muestra las variantes del modelo MLP, entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka ReliefFAttributeEval, que mejores resultados obtuvo; de un total de 25 variantes desarrolladas, las cuales se presentan en el anexo 18. Nótese que la Variante 133 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 136 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 40:** Variantes con mejores resultados del modelo MLP entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka ReliefFAttributeEval.

Variante	Algoritmo de entrenamiento	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>133</b>	entrenamiento de retropropagación con regularización bayesiana	5	0.066869	0.13777
<b>136</b>	entrenamiento de retropropagación del gradiente conjugado con reinicio Powell-Beale	1	0.10401	0.10592

#### 3.2.2.3.2 Modelo FFNN

La tabla 41 muestra la variante del modelo FFNN, entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka ReliefFAttributeEval,

que mejores resultados obtuvo; de un total de 10 variantes desarrolladas, las cuales se presentan en el anexo 19.

**Tabla 41:** Variante con mejores resultados del modelo FFNN entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval.

Variante	Arquitectura	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>58</b>	FFNN en cascada	5	0.087758	0.089404

### 3.2.2.3.3 Modelo LRN

La tabla 42 muestra los resultados obtenidos de cada una de las variantes del modelo LRN, entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval. Nótese que la Variante 27 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 26 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 42:** Resultados de las variantes del modelo LRN entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>26</b>	<b>1</b>	<b>0.027558</b>	<b>0.17821</b>
<b>27</b>	<b>3</b>	<b>0.018662</b>	<b>0.21885</b>
<b>28</b>	5	0.02196	0.20783
<b>29</b>	11	0.040203	0.18237
<b>30</b>	17	0.032314	0.18626

### 3.2.2.3.4 Modelo GRNN

La tabla 43 muestra los resultados obtenidos de cada una de las variantes del modelo GRNN, entrenado con el dataset de temporada baja con la selección de atributos

mediante el evaluador de Weka ReliefFAtributeEval. Nótese que la Variante 26 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 29 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 43:** Resultados de las variantes del modelo GRNN entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>26</b>	<b>1</b>	<b>5.2469e-08</b>	<b>0.10261</b>
<b>27</b>	3	1.2859e-07	0.098157
<b>28</b>	5	9.8224e-08	0.096273
<b>29</b>	<b>11</b>	<b>1.3623e-07</b>	<b>0.096225</b>
<b>30</b>	17	1.4626e-07	0.099163

### 3.2.2.3.5 Modelo NARX

La tabla 44 muestra los resultados obtenidos de cada una de las variantes del modelo NARX, entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval. Nótese que la Variante 26 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 27 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 44:** Resultados de las variantes del modelo NARX entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>26</b>	<b>1</b>	<b>0.073826</b>	<b>0.16863</b>
<b>27</b>	<b>3</b>	<b>0.093953</b>	<b>0.1389</b>
<b>28</b>	5	0.081799	0.1477
<b>29</b>	11	0.080466	0.15163
<b>30</b>	17	0.08669	0.14959

### 3.2.2.3.6 Modelo DT

La tabla 45 muestra los resultados obtenidos de cada una de las variantes del modelo DT, entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka ReliefFAttributeEval. Nótese que la Variante 30 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 26 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 45:** Resultados de las variantes del modelo DT entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka ReliefFAttributeEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>26</b>	<b>1</b>	<b>0.031426</b>	<b>0.033936</b>
<b>27</b>	3	0.026914	0.070328
<b>28</b>	5	0.024492	0.084821
<b>29</b>	11	0.02484	0.038952
<b>30</b>	<b>17</b>	<b>0.022142</b>	<b>0.046679</b>

### 3.2.2.3.7 Modelo SVM

La tabla 46 muestra los resultados obtenidos de cada una de las variantes del modelo SVM, entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka ReliefFAttributeEval. Nótese que la Variante 29 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 30 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 46:** Resultados de las variantes del modelo SVM entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
26	1	0.021297	0.050329
27	3	0.017629	0.064543
28	5	0.019246	0.08624
29	11	0.017566	0.052369
30	17	0.01923	0.041362

### 3.2.2.4 Conclusiones del epígrafe 3.2.2

Luego de aplicado el criterio RMSE a cada una de las variantes de los modelos, entrenados con los 3 datasets de temporada baja usados, se le obtiene la medida del desempeño de cada uno de estas variantes y se transita a seleccionar a los que mejor criterio RMSE tienen, los cuales se presentan en la tabla 47:

**Tabla 47:** Mejor resultado por cada dataset de la temporada baja.

Filtro de selección de atributos	Modelo	Variante	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
Ninguno	DT	16	0.031426	0.033936
CfsSubsetEval	SVM	25	0.0094611	0.044869
ReliefFAtributeEval	DT	26	0.031426	0.033936

## 3.2.3 Con el dataset del año completo

### 3.2.3.1 Haciendo uso del dataset completo

#### 3.2.3.1.1 Modelo MLP

La tabla 48 muestra las variantes del modelo MLP, entrenado con el dataset del año completo sin selección de atributos, que mejores resultados obtuvo; de un total de 25 variantes desarrolladas, las cuales se presentan en el anexo 20. Nótese que la Variante

159 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 164 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 48:** Variantes con mejores resultados del modelo MLP entrenado con el dataset del año completo sin selección de atributos.

Variante	Algoritmo de entrenamiento	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>159</b>	entrenamiento de retropropagación con regularización bayesiana	11	0.068156	0.11828
<b>164</b>	entrenamiento de retropropagación del gradiente conjugado con reinicio Powell-Beale	11	0.093951	0.092786

### 3.2.3.1.2 Modelo FFNN

La tabla 49 muestra la variante del modelo FFNN, entrenado con el dataset del año completo sin selección de atributos, que mejores resultados obtuvo; de un total de 10 variantes desarrolladas, las cuales se presentan en el anexo 21.

**Tabla 49:** Variante con mejores resultados del modelo FFNN entrenado con el dataset del año completo sin selección de atributos.

Variante	Arquitectura	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>66</b>	FFNN en cascada	1	0.087442	0.086437

### 3.2.3.1.3 Modelo LRN

La tabla 50 muestra los resultados obtenidos de cada una de las variantes del modelo LRN, entrenado con el dataset del año completo. Nótese que la Variante 33 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado

dataset, en la etapa de entrenamiento; mas es la Variante 31 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 50:** Resultados de las variantes del modelo LRN entrenado con el dataset del año completo sin selección de atributos.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>31</b>	<b>1</b>	<b>0.059249</b>	<b>0.16156</b>
<b>32</b>	3	0.039328	0.16237
<b>33</b>	<b>5</b>	<b>0.03667</b>	<b>0.16765</b>
<b>34</b>	11	0.047617	0.17518
<b>35</b>	17	0.042407	0.20813

#### 3.2.3.1.4 Modelo GRNN

La tabla 51 muestra los resultados obtenidos de cada una de las variantes del modelo GRNN, entrenado con el dataset del año completo. Nótese que la Variante 31 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 32 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 51:** Resultados de las variantes del modelo GRNN entrenado con el dataset del año completo sin selección de atributos.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>31</b>	<b>1</b>	<b>1.413e-07</b>	<b>0.093157</b>
<b>32</b>	<b>3</b>	<b>2.7169e-07</b>	<b>0.092237</b>
<b>33</b>	5	3.589e-07	0.092474
<b>34</b>	11	2.8667e-07	0.094837
<b>35</b>	17	8.23e-07	0.096423

### 3.2.3.1.5 Modelo NARX

La tabla 52 muestra los resultados obtenidos de cada una de las variantes del modelo NARX, entrenado con el dataset del año completo. Nótese que la Variante 35 señalada en azul, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset.

**Tabla 52:** Resultados de las variantes del modelo NARX entrenado con el dataset del año completo sin selección de atributos.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>31</b>	1	0.077833	0.12545
<b>32</b>	3	0.076516	0.13576
<b>33</b>	5	0.079328	0.14324
<b>34</b>	11	0.080926	0.12958
<b>35</b>	<b>17</b>	<b>0.076</b>	<b>0.12249</b>

### 3.2.3.1.6 Modelo DT

La tabla 53 muestra los resultados obtenidos de cada una de las variantes del modelo DT, entrenado con el dataset del año completo. Nótese que la Variante 32 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 31 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 53:** Resultados de las variantes del modelo DT entrenado con el dataset del año completo sin selección de atributos.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>31</b>	<b>1</b>	<b>0.012189</b>	<b>0.037475</b>
<b>32</b>	<b>3</b>	<b>0.010039</b>	<b>0.062369</b>
<b>33</b>	5	0.011246	0.073507
<b>34</b>	11	0.0132	0.0713
<b>35</b>	17	0.014082	0.052122

### 3.2.3.1.7 Modelo SVM

La tabla 54 muestra los resultados obtenidos de cada una de las variantes del modelo SVM, entrenado con el dataset del año completo. Nótese que la Variante 33 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 35 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 54:** Resultados de las variantes del modelo SVM entrenado con el dataset del año completo sin selección de atributos.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>31</b>	1	0.12857	0.095286
<b>32</b>	3	0.1296	0.12055
<b>33</b>	<b>5</b>	<b>0.12828</b>	<b>0.12003</b>
<b>34</b>	11	0.12935	0.10836
<b>35</b>	<b>17</b>	<b>0.12869</b>	<b>0.074403</b>

### 3.2.3.2 Haciendo uso del dataset luego de haber hecho la selección de atributos mediante el evaluador de Weka CfsSubsetEval

#### 3.2.3.2.1 Modelo MLP

La tabla 55 muestra las variantes del modelo MLP, entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka CfsSubsetEval, que mejores resultados obtuvo; de un total de 25 variantes desarrolladas, las cuales se presentan en el anexo 22. Nótese que la Variante 181 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 177 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 55:** Variantes con mejores resultados del modelo MLP entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Algoritmo de entrenamiento	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
177	entrenamiento de retropropagación BFGS quasi-Newton	3	0.11385	0.11301
181	entrenamiento de retropropagación con regularización bayesiana	1	0.11356	0.11401

### 3.2.3.2.2 Modelo FFNN

La tabla 56 muestra la variante del modelo FFNN, entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka CfsSubsetEval, que mejores resultados obtuvo; de un total de 10 variantes desarrolladas, las cuales se presentan en el anexo 23.

**Tabla 56:** Variante con mejores resultados del modelo FFNN entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Arquitectura	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
74	FFNN	11	0.10968	0.10876

### 3.2.3.2.3 Modelo LRN

La tabla 57 muestra los resultados obtenidos de cada una de las variantes del modelo LRN, entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka CfsSubsetEval. Nótese que la Variante 37 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 36 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 57:** Resultados de las variantes del modelo LRN entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>36</b>	<b>1</b>	<b>0.084174</b>	<b>0.14348</b>
<b>37</b>	<b>3</b>	<b>0.06933</b>	<b>0.15739</b>
<b>38</b>	5	0.086516	0.14733
<b>39</b>	11	0.075873	0.14577
<b>40</b>	17	0.077921	0.15404

#### 3.2.3.2.4 Modelo GRNN

La tabla 58 muestra los resultados obtenidos de cada una de las variantes del modelo GRNN, entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka CfsSubsetEval. Nótese que la Variante 38 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 36 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 58:** Resultados de las variantes del modelo GRNN entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>36</b>	<b>1</b>	<b>0.1101</b>	<b>0.11523</b>
<b>37</b>	3	0.10956	0.1166
<b>38</b>	<b>5</b>	<b>0.10942</b>	<b>0.11873</b>
<b>39</b>	11	0.11037	0.11657
<b>40</b>	17	0.10984	0.11668

#### 3.2.3.2.5 Modelo NARX

La tabla 59 muestra los resultados obtenidos de cada una de las variantes del modelo NARX, entrenado con el dataset del año completo con la selección de atributos mediante

el evaluador de Weka CfsSubsetEval. Nótese que la Variante 38 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 37 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 59:** Resultados de las variantes del modelo NARX entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>36</b>	1	0.082005	0.12414
<b>37</b>	<b>3</b>	<b>0.082491</b>	<b>0.12023</b>
<b>38</b>	<b>5</b>	<b>0.076227</b>	<b>0.13408</b>
<b>39</b>	11	0.083464	0.12773
<b>40</b>	17	0.079592	0.13515

### 3.2.3.2.6 Modelo DT

La tabla 60 muestra los resultados obtenidos de cada una de las variantes del modelo DT, entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka CfsSubsetEval. Nótese que la Variante 38 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 36 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 60:** Resultados de las variantes del modelo DT entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>36</b>	<b>1</b>	<b>0.024499</b>	<b>0.04109</b>
<b>37</b>	3	0.012325	0.045702
<b>38</b>	<b>5</b>	<b>0.0073972</b>	<b>0.074272</b>
<b>39</b>	11	0.032157	0.050461
<b>40</b>	17	0.0306	0.071979

### 3.2.3.2.7 Modelo SVM

La tabla 61 muestra los resultados obtenidos de cada una de las variantes del modelo SVM, entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka CfsSubsetEval. Nótese que la Variante 36 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 40 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 61:** Resultados de las variantes del modelo SVM entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>36</b>	<b>1</b>	<b>0.074147</b>	<b>0.072482</b>
<b>37</b>	3	0.075269	0.08871
<b>38</b>	5	0.076818	0.098464
<b>39</b>	11	0.07732	0.078754
<b>40</b>	<b>17</b>	<b>0.076076</b>	<b>0.068361</b>

### 3.2.3.3 Haciendo uso del dataset luego de haber hecho la selección de atributos mediante el evaluador de Weka ReliefFAttributeEval

#### 3.2.3.3.1 Modelo MLP

La tabla 62 muestra las variantes del modelo MLP, entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka ReliefFAttributeEval, que mejores resultados obtuvo; de un total de 25 variantes desarrolladas, las cuales se presentan en el anexo 24. Nótese que la Variante 209 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 214 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 62:** Variantes con mejores resultados del modelo MLP entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka ReliefFAttributeEval.

Variante	Algoritmo de entrenamiento	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>209</b>	entrenamiento de retropropagación con regularización bayesiana	11	0.068156	0.11828
<b>214</b>	entrenamiento de retropropagación del gradiente conjugado con reinicio Powell-Beale	11	0.093951	0.092786

### 3.2.3.3.2 Modelo FFNN

La tabla 63 muestra la variante del modelo FFNN, entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka ReliefFAttributeEval, que mejores resultados obtuvo; de un total de 10 variantes desarrolladas, las cuales se presentan en el anexo 25.

**Tabla 63:** Variante con mejores resultados del modelo FFNN entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka ReliefFAttributeEval.

Variante	Arquitectura	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>86</b>	FFNN en cascada	1	0.087442	0.086437

### 3.2.3.3.3 Modelo LRN

La tabla 64 muestra los resultados obtenidos de cada una de las variantes del modelo LRN, entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka ReliefFAttributeEval. Nótese que la Variante 43 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en

la etapa de entrenamiento; mas es la Variante 41 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 64:** Resultados de las variantes del modelo LRN entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>41</b>	<b>1</b>	<b>0.059249</b>	<b>0.16156</b>
<b>42</b>	3	0.039328	0.16237
<b>43</b>	<b>5</b>	<b>0.03667</b>	<b>0.16765</b>
<b>44</b>	11	0.047617	0.17518
<b>45</b>	17	0.042407	0.20813

#### 3.2.3.3.4 Modelo GRNN

La tabla 65 muestra los resultados obtenidos de cada una de las variantes del modelo GRNN, entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval. Nótese que la Variante 41 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 43 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 65:** Resultados de las variantes del modelo GRNN entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>41</b>	<b>1</b>	<b>1.413e-07</b>	<b>0.093157</b>
<b>42</b>	<b>3</b>	<b>2.7169e-07</b>	<b>0.092237</b>
<b>43</b>	5	3.589e-07	0.092474
<b>44</b>	11	2.8667e-07	0.094837
<b>45</b>	17	8.23e-07	0.096423

### 3.2.3.3.5 Modelo NARX

La tabla 66 muestra los resultados obtenidos de cada una de las variantes del modelo NARX, entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval. Nótese que la Variante 45 señalada en azul, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset.

**Tabla 66:** Resultados de las variantes del modelo NARX entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
41	1	0.077833	0.12545
42	3	0.076516	0.13576
43	5	0.079328	0.14324
44	11	0.080926	0.12958
45	17	0.076	0.12249

### 3.2.3.3.6 Modelo DT

La tabla 67 muestra los resultados obtenidos de cada una de las variantes del modelo DT, entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval. Nótese que la Variante 42 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 41 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 67:** Resultados de las variantes del modelo DT entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
41	1	0.012189	0.037475
42	3	0.010039	0.062369
43	5	0.011246	0.073507

<b>44</b>	11	0.0132	0.0713
<b>45</b>	17	0.014082	0.052122

### 3.2.3.3.7 Modelo SVM

La tabla 68 muestra los resultados obtenidos de cada una de las variantes del modelo SVM, entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval. Nótese que la Variante 43 señalada en verde, muestra el mejor resultado para dicho modelo, haciendo uso del mencionado dataset, en la etapa de entrenamiento; mas es la Variante 45 la que muestra un mejor resultado para la etapa de prueba.

**Tabla 68:** Resultados de las variantes del modelo SVM entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval.

Variante	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>41</b>	1	0.12857	0.095286
<b>42</b>	3	0.1296	0.12055
<b>43</b>	<b>5</b>	<b>0.12828</b>	<b>0.12003</b>
<b>44</b>	11	0.12935	0.10836
<b>45</b>	<b>17</b>	<b>0.12869</b>	<b>0.074403</b>

### 3.2.3.4 Conclusiones del epígrafe 3.2.3

Luego de aplicado el criterio RMSE a cada una de las variantes de los modelos, entrenados con los 3 datasets del año completo usados, se le obtiene la medida del desempeño de cada uno de estas variantes y se transita a seleccionar a los que mejor criterio RMSE tienen, los cuales se presentan en la tabla 69:

**Tabla 69:** Mejor resultado por cada dataset del año completo.

Filtro de selección de atributos	Modelo	Variante	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
----------------------------------	--------	----------	-----------------------------------	----------------------------

<b>Ninguno</b>	DT	31	0.012189	0.037475
<b>CfsSubsetEval</b>	DT	36	0.024499	0.04109
<b>ReliefFAtributeEval</b>	DT	41	0.012189	0.037475

### 3.3 Mejor modelo por dataset

Luego de aplicado el criterio RMSE a cada una de las variantes de los modelos, se le obtiene la medida del desempeño de cada uno de estas variantes y se transita a seleccionar a los que mejor criterio RMSE tienen por cada dataset, los cuales se presentan en la tabla 70.

**Tabla 70:** Mejor resultado por cada dataset

Temporada del dataset	Filtro de selección de atributos	Modelo	Variante	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>Temporada alta</b>	Ninguno	DT	2	0.018633	0.060149
	CfsSubsetEval	DT	8	0.041286	0.086782
	ReliefFAtributeEval	DT	12	0.018634	0.060149
<b>Temporada baja</b>	Ninguno	DT	16	0.031426	0.033936
	CfsSubsetEval	SVM	25	0.0094611	0.044869
	ReliefFAtributeEval	DT	26	0.031426	0.033936
<b>Año completo</b>	Ninguno	DT	31	0.012189	0.037475
	CfsSubsetEval	DT	36	0.024499	0.04109
	ReliefFAtributeEval	DT	41	0.012189	0.037475

## **3.4 Modelos de minería de datos propuestos para la predicción de energía eléctrica de una edificación hotelera**

### **3.4.1 Mejores modelos según el criterio RMSE**

El resultado que se selecciona como el mejor modelo por cada temporada del año, para la predicción de energía eléctrica de una edificación hotelera, según el criterio RMSE, presentan las siguientes características:

- Para la temporada alta:

Modelo con arquitectura de árbol de decisión, en su variante 2; creado, entrenado y probado con una semilla de generación aleatoria igual a 3 y el dataset de temporada alta sin selección de atributos. En los demás aspectos referentes a las configuraciones del árbol de decisión se dejaron por según vienen por defecto con MATLAB.

El código resultante de la implementación, en MATLAB, del modelo seleccionado aparece reflejado en el Anexo 26.

- Para la temporada baja:

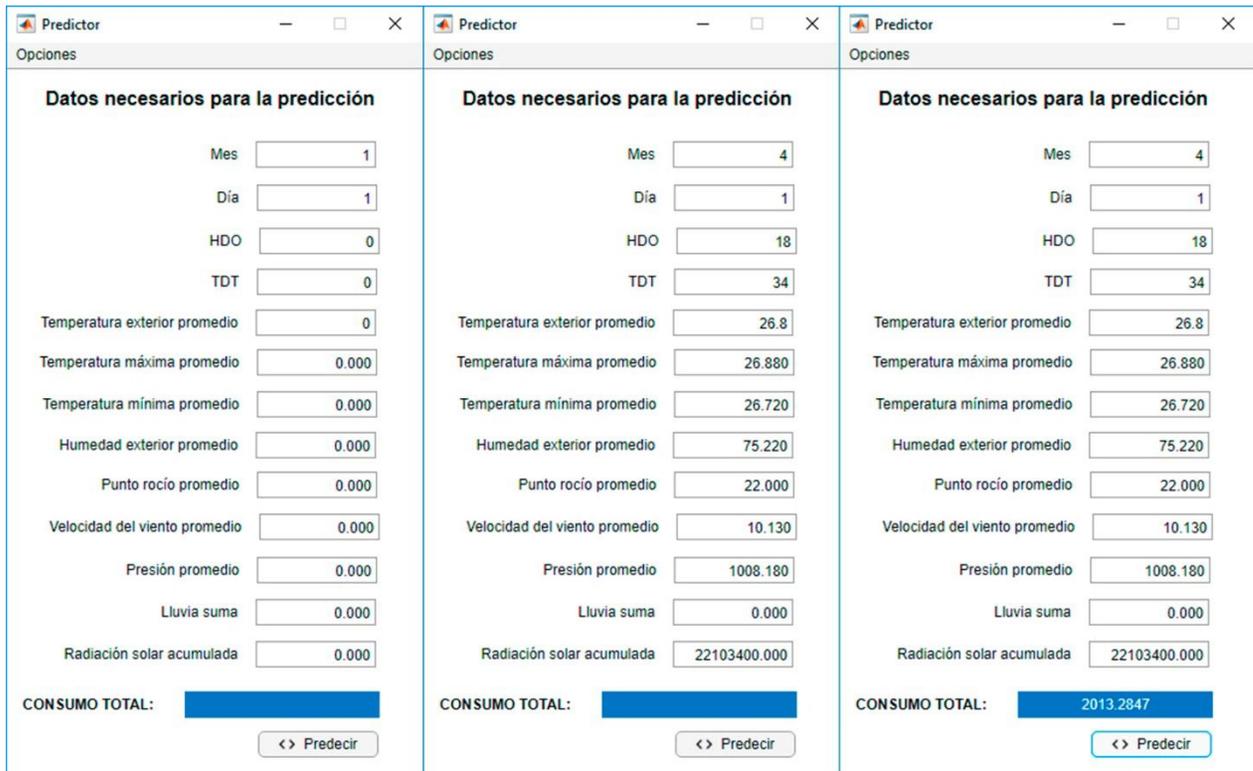
Modelo con arquitectura de árbol de decisión, en su variante 16; creado, entrenado y probado con una semilla de generación aleatoria igual a 1 y el dataset de temporada baja sin selección de atributos. En los demás aspectos referentes a las configuraciones del árbol de decisión se dejaron por según vienen por defecto con MATLAB.

El código resultante de la implementación, en MATLAB, del modelo seleccionado aparece reflejado en el Anexo 27.

## **3.5 Interfaz gráfica**

Se implementó en MATLAB una interfaz gráfica (Graphical User Interface; GUI) mediante la cual los usuarios podrán interactuar con los modelos predictivos propuestos. La GUI permite al usuario introducir de manera sencilla los valores de las variables necesarias

para la predicción, y realizar tal predicción. El desarrollo de esta interfaz gráfica se hizo en la herramienta proporcionada por MATLAB llamada MATLAB App Designer («MATLAB App Designer», 2019), y fue además compilada gracias al compilador de MATLAB («MATLAB Compiler», 2019). Esto tiene como desventaja que el computador donde la GUI vaya a ser utilizada necesita el MATLAB Component Runtime (MCR) («MATLAB Runtime», 2019) para poder funcionar. En la figura 9 se muestra la interfaz gráfica desarrollada durante el proceso de predicción.



**Fig. 9** Interfaz gráfica desarrollada en MATLAB para la interacción con los modelos seleccionados.

## Conclusiones del capítulo

- El criterio utilizado para medir el desempeño de los modelos fue el RMSE.
- Los modelos seleccionados para la predicción de la energía eléctrica de una edificación hotelera, son el DT en su variante 2; creado, entrenado y probado con una semilla de generación aleatoria igual a 3 y el dataset de temporada alta sin selección

de atributos, y el DT en su variante 16; creado, entrenado y probado con una semilla de generación aleatoria igual a 1 y el dataset de temporada baja sin selección de atributos; respectivamente.

- Se desarrolló una interfaz gráfica en MATLAB que le brinda a los especialistas en energía la posibilidad de interactuar con los modelos.

## Conclusiones

- La metodología de minería de datos seleccionada para el desarrollo de la experimentación es KDD.
- Las técnicas de minería de datos más usadas para la predicción del consumo de energía eléctrica en edificaciones hoteleras son: ANN, SVM y los DT.
- Se experimentaron con 225 variantes del MLP, 90 variantes del FFNN, 45 variantes del LRN, 45 variantes del GRNN, 45 variantes del NARX, 45 variantes del DT y 45 variantes del SVM.
- Se proponen dos modelos que brindan a los especialistas en energía una herramienta de apoyo en la toma de decisiones:
  - Para la temporada alta: el modelo DT (creado, entrenado y probado con una semilla de generación aleatoria igual a 3 y el dataset de temporada alta sin selección de atributos) con una efectividad de 93.99 %.
  - Para la temporada baja: el modelo DT (creado, entrenado y probado con una semilla de generación aleatoria igual a 1 y el dataset de temporada baja sin selección de atributos) con una efectividad de 96.61 %.

## Recomendaciones

1. Recolectar una mayor cantidad de datos y realizar una nueva experimentación con el objetivo de mejorar la efectividad.
2. Experimentar con otras técnicas de minería de datos no utilizadas en esta investigación.
3. Desarrollar una aplicación informática que permita a los energéticos del hotel predecir el consumo de energía para un día específico, conocidas las variables climatológicas y de ocupación.

## Referencias Bibliográficas

- A. Sierra, D., Manzi, D., M. Brentan, B., & Luvizotto Jr., E. (2017). Metodologías para análisis de error de previsión de demanda para identificar eventos anómalos en redes de abastecimiento de agua.
- América Latina es líder en el ahorro energético mundial. (2017.). Recuperado a partir de <https://www.telesurtv.net/news/America-Latina-es-lider-en-el-ahorro-energetico-mundial-20171019-0048.html>
- Anne Koehler. (s. f.). Recuperado a partir de <http://miamioh.edu/fsb/directory/?up=/directory/koehleab>
- Arencibia, Y. (2018). *Redes Neuronales Recurrentes*.
- Attanasio, A., & Giorgi, J. (s. f.). *Energía solar, la consolidación de la fuente alternativa*. Recuperado a partir de <https://www.elperiodico.com/es/especiales/connecting-africa-desarrollo/energia-solar.html>.
- Bergmeir, C., & Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation.
- Betancourt, G. (2010). LAS MÁQUINAS DE SOPORTE VECTORIAL (SVMs).
- Cabrera Gorrin, O., & Pérez Dorta, Y. (2019). Reflexiones sobre el consumo energético en el sector hotelero cubano.
- Cadena hotelera HORIZONTES. (s. f.). Recuperado a partir de <https://www.cubanacan.cu/es/hotelbychain/horizontes>
- Camejo Corona, J. (2012, septiembre). *Introducción a la Minería de datos*.
- Cárdenes Almeida, R. (s. f.). Inteligencia Artificial.
- Carmona Suárez, E. J. (2013). Tutorial sobre Máquinas de Vectores Soporte (SVM).
- Casali, A. (s. f.). ¿Qué es la Inteligencia Artificial?
- CfsSubsetEval. (s. f.). Recuperado a partir de <http://weka.sourceforge.net/doc.stable/weka/attributeSelection/CfsSubsetEval.html>
- Chalal, M., Benachir, M., White, M., & Shrahily, R. (2016). Energy planning and forecasting approaches for supporting physical improvement strategies in the building sector: a review.
- Chen, F., Zhigao, Z., Aiwen, L., Jiqiang, N., Wenmin, Q., & Zhong, Y. (2019). *Evaluation of Direct Horizontal Irradiance in China Using a Physically-Based Model and Machine Learning Methods*. Recuperado a partir de [https://www.researchgate.net/publication/330112239\\_Evaluation\\_of\\_Direct\\_Horizontal\\_Irradiance\\_in\\_China\\_Using\\_a\\_Physically-Based\\_Model\\_and\\_Machine\\_Learning\\_Methods/fulltext/5c2e2221a6fdccd6b58f7e83/330112239\\_Evaluation\\_of\\_Direct\\_Horizontal\\_Irradiance\\_in\\_China\\_Using\\_a\\_Physically-Based\\_Model\\_and\\_Machine\\_Learning\\_Methods.pdf?origin=publication\\_detail](https://www.researchgate.net/publication/330112239_Evaluation_of_Direct_Horizontal_Irradiance_in_China_Using_a_Physically-Based_Model_and_Machine_Learning_Methods/fulltext/5c2e2221a6fdccd6b58f7e83/330112239_Evaluation_of_Direct_Horizontal_Irradiance_in_China_Using_a_Physically-Based_Model_and_Machine_Learning_Methods.pdf?origin=publication_detail)
- Cheng, F., Fu, X., & Shengwei, W. (2014). Development of prediction models for next-day building energy consumption and peak power demand using data mining techniques.
- Christoph Bergmeir. (s. f.). Recuperado a partir de <https://research.monash.edu/en/persons/christoph-bergmeir>
- Colmenares, G. (2009, diciembre 3). MÁQUINA DE VECTORES DE SOPORTE.
- Conjugate Gradient Algorithms. (s. f.). Recuperado a partir de <http://matlab.izmiran.ru/help/toolbox/nnet/backpr59.html>
- CRISP-DM. (s. f.). Recuperado a partir de <https://www.sv-europe.com/crisp-dm-methodology/>
- de Armas, N. (s. f.). *Energía y su uso eficiente, una preocupación de América Latina*. Recuperado a partir de <http://www.fao.org/in-action/agronoticias/detail/es/c/512060/>
- Demuth, H., Beale, M., & Hagan, M. (2010). Neural Network Toolbox 7 User's Guide MatLab.

- Diego García Morate. (s. f.). MANUAL DE WEKA.
- Diuber Sicilia Camacho, Mario Herrera Prat, Miguel Pelayo García Fernández, & Ángel Espinosa Nápoles. (2011). Modelo matemático para estimar el consumo de energía eléctrica en instalaciones de la educación superior.
- Domingo Laino, L. (2015). Un Análisis de la Política Energética en Cuba.
- El Congreso de Colombia. (2001). Ley No. 697 de 2001.
- F., T. (2010). Introducción a las redes neuronales artificiales.
- Fumo, N. (2013). A review on the basics of building energy estimation.
- Gaiser, T., de Barros, I., Sereke, F., & Lange, F. M. (2010). Validation and reliability of the EPIC model to simulate maize production in small- holder farming systems in tropical sub-humid West Africa and semi- arid Brazil.
- General Regression Neural Network (GRNN). (s. f.).
- GRUPO RENDER INDUSTRIAL S.L. (2015, marzo 26). Estrategia Energética en la Unión Europea.
- GUSTAVO A. BETANCOURT. (s. f.). LAS MÁQUINAS DE SOPORTE VECTORIAL (SVMs).
- H., D., M., B., & M., H. (2007). Neural Network Toolbox 5 User's Guide MatLab.
- H., D., M., B., & M., H. (2017). *Neural Network Design*.
- H. Demuth, M. Beale, & M. Hagan. (2007). Neural Network Toolbox 5 User's Guide MatLab.
- Hai-xiang, Z., & Frédéric, M. (2012). A review on the prediction of building energy consumption.
- Hernández Orallo, J., Ramírez Quintana, J., & Ferri Ramírez, C. (2011). *Introducción a la Minería de Datos*. PEARSON EDUCACIÓN.
- J. Hyndman, R., & B. Koehler, A. (s. f.). Another look at measures of forecast accuracy.
- J., M. (2012). Introducción a las redes neuronales aplicadas.
- Jones, D. A. (s. f.). *Electrical engineering: the backbone of society*. Recuperado a partir de <http://dx.doi.org/10.1049%2Fip-a-3.1991.0001>
- José Manuel Benítez Sánchez. (2019). Recuperado a partir de <http://citic.ugr.es/personal/J.M.Benitez@decsai.ugr.es>
- Kadir, A., & Nora, M. E.-G. (2018). A review of data-driven building energy consumption prediction studies.
- Karamirad, M., Omid, M., Alimardani, R., Mousazadeh, H., & Heidari, S. N. (2013). ANN based simulation and experimental verification of analytical four- and five-parameters models of PV modules.
- Khaled Alsedrah, M. (2017). Artificial Intelligence.
- Kuldeep, S. (2016). AN INTRODUCTION TO ARTIFICIAL NEURAL NETWORK.
- L., F. (2005). *Entrenamiento de Redes Neuronales Basado en Algoritmos Evolutivos*. Universidad de Buenos Aires, Argentina.
- León Guzmán, E. (2012). Metodologías aplicadas al proceso de Minería de Datos.
- Li, X., & Wen, J. (2014). Review of building energy modeling for control and operation.
- Lighting Africa. (2018). Recuperado a partir de <https://www.lightingafrica.org/>
- Lior, R., & Oded, M. (2015). *DATA MINING WITH DECISION TREES Theory and Applications*. World Scientific Publishing Co. Pte. Ltd.
- Liu, H., Li, Y., & Q. Tian, H. (2010). A hybrid statistical method to predict wind speed and wind power.
- Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and Machine Learning forecasting methods: Concerns and ways forward.
- Marín Quemada, J. (2012). POLÍTICA ENERGÉTICA EN LA UE: El debate entre la timidez y el atrevimiento.

- Matlab. (2019). Recuperado a partir de <https://www.monografias.com/trabajos5/matlab/matlab.shtml>
- MATLAB App Designer. (2019). Recuperado a partir de <https://www.mathworks.com/products/matlab/app-designer.html>
- MATLAB Compiler. (2019). Recuperado a partir de <https://www.mathworks.com/products/compiler.html>
- MATLAB Runtime. (2019). Recuperado a partir de <https://es.mathworks.com/products/compiler/matlab-runtime.html>
- Molina González, A., Velarde Bedregal, H. R., E. Santiesteban Toca, C., & P. Monteagudo Yanes, J. (2017). Nuevos índices de consumo energético para hoteles tropicales.
- MORENO RODRÍGUEZ, A. (2009). *Desarrollo de una interfaz gráfica de redes neuronales usando Matlab*. UNIVERSIDAD CARLOS III MADRID. España.
- Nadaraya, E. A. (1964). On Estimating Regression.
- ONURE. (2017, noviembre). Eficiencia Energética. Actualidad y Proyecciones.
- Oropeza, C. (s. f.). capítulo 3: Redes Neuronales Recurrentes. Recuperado a partir de [http://catarina.udlap.mx/u\\_dl\\_a/tales/documentos/lem/oropeza\\_c\\_ca/capitulo3.pdf](http://catarina.udlap.mx/u_dl_a/tales/documentos/lem/oropeza_c_ca/capitulo3.pdf)
- Proyecto Reforma la ley 7447. UREE en Costa Rica. (2019). Recuperado a partir de <https://www.cepal.org/es/publicaciones/6320-proyecto-reforma-la-ley-7447-regulacion-uso-racional-la-energia-costa-rica>
- Ramos Niembro, G., Fiscal Escalante, R., Maqueda Zamora, M., Sada Gámiz, J., & Buitrón Sánchez, H. (2016). Variables que influyen en el consumo de energía eléctrica.
- ReliefFAttributeEval. (2018). Recuperado a partir de <http://weka.sourceforge.net/doc.stable/weka/attributeSelection/ReliefFAttributeEval.html>
- Revolución Energética en Cuba. (2015). En *Ecured*.
- Rob J Hyndman. (2019). Recuperado a partir de <https://robjhyndman.com/about/>
- ROQUE MONTALVO, I. L. (2016, noviembre 9). *ANÁLISIS COMPARATIVO DE TÉCNICAS DE MINERÍA DE DATOS PARA LA PREDICCIÓN DE VENTAS*.
- S., M., M., de A., A., B., J., G., & C., P. (s. f.). Inteligencia Artificial Aplicada a la Reducción del Consumo Energético de un Sistema de Climatización por agua Helada en un Hotel Turístico.
- Sánchez Borroto, Y., Melo Espinosa, E. A., Piloto Rodríguez, R., & Sierens, R. (2014). EL ESCENARIO ENERGÉTICO CUBANO.
- Sancho Caparrini, F. (s. f.). Aprendizaje Inductivo: Árboles de Decisión. Recuperado a partir de <http://www.cs.us.es/~fsancho/?e=104>
- Santiesteban Velázquez, A., Osvaldo Nápoles González, J., Rodríguez Peña, D., & Jaime García, D. (2014). Pronóstico de consumo de energía eléctrica usando redes neuronales artificiales.
- SEMMA. (2015). Recuperado a partir de <https://101.datascience.community/tag/semma/>
- Sicilia Camacho, D., Herrera Prat, M., García Fernández, M. P., & Espinosa Nápoles, Á. (2011). Modelo matemático para estimar el consumo de energía eléctrica en instalaciones de la educación superior.
- Sitio oficial de descarga de MATLAB en su versión r2018a. (2019). Recuperado a partir de [https://www.mathworks.com/products/new\\_products/release2018a.html](https://www.mathworks.com/products/new_products/release2018a.html)
- Sitio oficial de GPL. (2019). Recuperado a partir de <http://www.gnu.org/copyleft/gpl.html>
- Sitio oficial de Java. (2019). Recuperado a partir de <https://www.java.com/>
- Sitio oficial de KDD. (2019). Recuperado a partir de <https://www.kdd.org/>
- Sitio oficial de la Agencia Chilena de Eficiencia Energética. (2019). Recuperado a partir de <http://old.acee.cl/>
- Sitio oficial de la Agencia Internacional de Energía. (2019). Recuperado a partir de <https://www.iea.org/>

- Sitio oficial de la cadena hotelera CUBANACAN. (2019). Recuperado a partir de <https://www.cubanacan.cu/>
- Sitio oficial de la cadena hotelera GRAN CARIBE. (2019). Recuperado a partir de <https://www.gran-caribe.cu/>
- Sitio oficial de la Unión Cuba-Petróleo. (s. f.-a). Recuperado a partir de <https://www.cupet.cu/>
- Sitio oficial de la Unión Eléctrica de Cuba. (2018). Recuperado a partir de <https://www.unionelectrica.cu/>
- Sitio oficial de la Universidad de Waikato en Nueva Zelanda. (2019). Recuperado a partir de <https://www.waikato.ac.nz/>
- Sitio oficial de MATLAB. (s. f.-b). Recuperado a partir de <https://www.mathworks.com/>
- Sitio oficial de Weka. (s. f.-c). Recuperado a partir de [www.cs.waikato.ac.nz/ml/weka](http://www.cs.waikato.ac.nz/ml/weka)
- Sitio oficial del Hotel Meliá Habana de la Ciudad de la Habana. (2019). Recuperado a partir de <https://www.melia.com/en/hotels/cuba/havana/melia-habana/index.htm>
- Sitio oficial del Ministerio de Energía y Minas de Cuba. (2019). Recuperado a partir de <https://www.minem.gob.cu/>
- Sitio oficial del Ministerio de la Construcción de Cuba. (2019). Recuperado a partir de [www.micons.gob.cu/](http://www.micons.gob.cu/)
- Sitio oficial del Ministerio de la Industria Básica de Cuba. (2019). Recuperado a partir de [www.mindus.gob.cu/](http://www.mindus.gob.cu/)
- Toledo, V. (2017). *METODOLOGÍAS DE MINERÍA DE DATOS*.
- V. Acosta, A. (2011). Modelo para la Predicción Energética de una Instalación Hotelera.
- Varela Arregocés, E., & Campbells Sánchez, E. (2011). *Redes Neuronales Artificiales: Una Revisión del Estado del Arte, Aplicaciones Y Tendencias Futuras*.
- Vásquez López, J. P. (2014). *RED NEURONAL FEEDFORWARD COMO ESTIMADOR DE PATRONES DE CORRIENTES EN EL INTERIOR DEL PUERTO DE MANZANILLO SUJETO A LA ACCIÓN DE TSUNAMIS*.
- Vélez Correa, J., & Nieto Figueroa, P. (2016). *VALIDACIÓN DE MEDIDAS DE EVALUACIÓN PARA EL PRONÓSTICO DE LA TASA DE CAMBIO EN COLOMBIA*. Colegio de Estudios Superiores de Administración –CESA–.
- Visanzay, G. (s. f.). Identificación de un proceso de secado utilizando Redes Neuronales Dinámicas.
- Vladimir Vapnik. (2019). Recuperado a partir de <https://datascience.columbia.edu/vladimir-vapnik>
- Watson, G. S. (1964). Smooth regression analysis.
- Z., W., & RS, S. (2016). A review of artificial intelligence based building energy use prediction: Contrasting the capabilities of single and ensemble prediction models.
- Zhengwei, L., Yanmin, H., & Peng, X. (2014). Methods for benchmarking building energy consumption against its past or intended performance: An overview.

## Glosario de términos

MW: Megavatio o Megawatt.

BFGS: Actualización Broyden, Fletcher, Goldfarb, y Shanno del método quasi-Newton.

Kw: Kilovatio o kilowatt.

m/s: Metro por segundo.

°C: Grados Celsius, (históricamente conocido como centígrado).

KPa: Kilopascal.

mm: Milímetro.

W/m<sup>2</sup>: Watt por metro al cuadrado.

## Anexos

### Anexo 1. Variantes del MLP experimentadas en MATLAB

Variante	Temporada del dataset	Filtro de selección de atributos	Función de entrenamiento	Semilla de generación aleatoria
1	Temporada alta	Ninguno	entrenamiento de retropropagación BFGS quasi-Newton	1
2				3
3				5
4				11
5				17
6			entrenamiento de retropropagación con regularización bayesiana	1
7				3
8				5
9				11
10				17
11			entrenamiento de retropropagación del gradiente conjugado con reinicio Powell-Beale	1
12				3
13				5
14				11
15				17
16			entrenamiento de retropropagación del gradiente conjugado con actualización Fletcher-Reeves	1
17				3
18				5
19				11
20				17
21			entrenamiento de retropropagación del gradiente descendente	1
22				3
23				5
24				11
25				17
26		CfsSubsetEval	entrenamiento de retropropagación BFGS quasi-Newton	1
27				3
28				5
29				11
30				17
31			entrenamiento de retropropagación con regularización bayesiana	1
32				3
33				5
34				11
35				17

36			entrenamiento de retropropagación del gradiente conjugado con reinicio Powell-Beale	1
37				3
38				5
39				11
40				17
41			entrenamiento de retropropagación del gradiente conjugado con actualización Fletcher-Reeves	1
42				3
43				5
44				11
45				17
46			entrenamiento de retropropagación del gradiente descendente	1
47				3
48				5
49				11
50				17
51		ReliefFAtributeEval	entrenamiento de retropropagación BFGS quasi-Newton	1
52				3
53				5
54				11
55				17
56			entrenamiento de retropropagación con regularización bayesiana	1
57				3
58				5
59				11
60				17
61			entrenamiento de retropropagación del gradiente conjugado con reinicio Powell-Beale	1
62				3
63				5
64				11
65				17
66			entrenamiento de retropropagación del gradiente conjugado con actualización Fletcher-Reeves	1
67				3
68				5
69				11
70				17
71			entrenamiento de retropropagación del gradiente descendente	1
72				3
73				5
74				11
75				17
76	Temporada baja	Ninguno		1
77				3

78			entrenamiento de retropropagación BFGS quasi-Newton	5
79				11
80				17
81			entrenamiento de retropropagación con regularización bayesiana	1
82				3
83				5
84				11
85				17
86			entrenamiento de retropropagación del gradiente conjugado con reinicio Powell-Beale	1
87				3
88				5
89				11
90				17
91			entrenamiento de retropropagación del gradiente conjugado con actualización Fletcher-Reeves	1
92				3
93				5
94				11
95				17
96			entrenamiento de retropropagación del gradiente descendente	1
97				3
98				5
99				11
100				17
101		CfsSubsetEval	entrenamiento de retropropagación BFGS quasi-Newton	1
102				3
103				5
104				11
105				17
106			entrenamiento de retropropagación con regularización bayesiana	1
107				3
108				5
109				11
110				17
111			entrenamiento de retropropagación del gradiente conjugado con reinicio Powell-Beale	1
112				3
113				5
114				11
115				17
116			entrenamiento de retropropagación del gradiente conjugado con actualización Fletcher-Reeves	1
117				3
118				5
119				11
120				17

121			entrenamiento de retropropagación del gradiente descendente	1
122				3
123				5
124				11
125				17
126		ReliefFAtributeEval	entrenamiento de retropropagación BFGS quasi-Newton	1
127				3
128				5
129				11
130				17
131			entrenamiento de retropropagación con regularización bayesiana	1
132				3
133				5
134				11
135				17
136			entrenamiento de retropropagación del gradiente conjugado con reinicio Powell-Beale	1
137				3
138				5
139				11
140				17
141			entrenamiento de retropropagación del gradiente conjugado con actualización Fletcher-Reeves	1
142				3
143				5
144				11
145				17
146			entrenamiento de retropropagación del gradiente descendente	1
147				3
148				5
149				11
150				17
151	Año completo	Ninguno	entrenamiento de retropropagación BFGS quasi-Newton	1
152				3
153				5
154				11
155				17
156			entrenamiento de retropropagación con regularización bayesiana	1
157				3
158				5
159				11
160				17
161			entrenamiento de retropropagación del gradiente conjugado	1
162				3
163				5

164			con reinicio Powell-Beale	11
165				17
166			entrenamiento de retropropagación del gradiente conjugado con actualización Fletcher-Reeves	1
167				3
168				5
169				11
170				17
171			entrenamiento de retropropagación del gradiente descendente	1
172				3
173				5
174				11
175				17
176		CfsSubsetEval	entrenamiento de retropropagación BFGS quasi-Newton	1
177				3
178				5
179				11
180				17
181			entrenamiento de retropropagación con regularización bayesiana	1
182				3
183				5
184				11
185				17
186			entrenamiento de retropropagación del gradiente conjugado con reinicio Powell-Beale	1
187				3
188				5
189				11
190				17
191			entrenamiento de retropropagación del gradiente conjugado con actualización Fletcher-Reeves	1
192				3
193				5
194				11
195				17
196			entrenamiento de retropropagación del gradiente descendente	1
197				3
198				5
199				11
200				17
201		ReliefFAtributeEval	entrenamiento de retropropagación BFGS quasi-Newton	1
202				3
203				5
204				11
205				17
206				1

207			entrenamiento de retropropagación con regularización bayesiana	3
208				5
209				11
210				17
211			entrenamiento de retropropagación del gradiente conjugado con reinicio Powell-Beale	1
212				3
213				5
214				11
215				17
216			entrenamiento de retropropagación del gradiente conjugado con actualización Fletcher-Reeves	1
217				3
218				5
219				11
220				17
221			entrenamiento de retropropagación del gradiente descendente	1
222				3
223				5
224				11
225				17

## Anexo 2. Variantes del FFNN experimentadas en MATLAB

Variante	Arquitectura	Temporada del dataset	Filtro de selección de atributos	Semilla de generación aleatoria
1	FFNN	Temporada alta	Ninguno	1
2				3
3				5
4				11
5				17
6			CfsSubsetEval	1
7				3
8				5
9				11
10				17
11			ReliefFAttributeEval	1
12				3
13				5
14				11
15				17
16		Temporada baja	Ninguno	1
17				3

18				5
19				11
20				17
21			CfsSubsetEval	1
22				3
23				5
24				11
25				17
26			ReliefFAttributeEval	1
27				3
28				5
29				11
30				17
31		Año completo	Ninguno	1
32				3
33				5
34				11
35				17
36			CfsSubsetEval	1
37				3
38				5
39				11
40				17
41			ReliefFAttributeEval	1
42				3
43				5
44				11
45				17
46	FFNN en cascada	Temporada alta	Ninguno	1
47				3
48				5
49				11
50				17
51			CfsSubsetEval	1
52				3
53				5
54				11
55				17
56			ReliefFAttributeEval	1
57				3
58				5
59				11
60				17

61		Temporada baja	Ninguno	1
62				3
63				5
64				11
65				17
66			CfsSubsetEval	1
67				3
68				5
69				11
70				17
71			ReliefFAttributeEval	1
72				3
73				5
74				11
75				17
76		Año completo	Ninguno	1
77				3
78				5
79				11
80				17
81			CfsSubsetEval	1
82				3
83				5
84				11
85				17
86			ReliefFAttributeEval	1
87				3
88				5
89				11
90				17

### Anexo 3. Variantes del LRN experimentadas en MATLAB

Variante	Temporada del dataset	Filtro de selección de atributos	Semilla de generación aleatoria
1	Temporada alta	Ninguno	1
2			3
3			5
4			11
5			17
6		CfsSubsetEval	1
7			3

8			5
9			11
10			17
11		ReliefFAtributeEval	1
12			3
13			5
14			11
15			17
16	Temporada baja	Ninguno	1
17			3
18			5
19			11
20			17
21		CfsSubsetEval	1
22			3
23			5
24			11
25			17
26		ReliefFAtributeEval	1
27			3
28			5
29			11
30			17
31	Año completo	Ninguno	1
32			3
33			5
34			11
35			17
36		CfsSubsetEval	1
37			3
38			5
39			11
40			17
41		ReliefFAtributeEval	1
42			3
43			5
44			11
45			17

#### Anexo 4. Variantes del GRNN experimentadas en MATLAB

Variante	Temporada del dataset	Filtro de selección de atributos	Semilla de generación aleatoria
1	Temporada alta	Ninguno	1
2			3
3			5
4			11
5			17
6		CfsSubsetEval	1
7			3
8			5
9			11
10			17
11		ReliefFAtributeEval	1
12			3
13			5
14			11
15			17
16	Temporada baja	Ninguno	1
17			3
18			5
19			11
20			17
21		CfsSubsetEval	1
22			3
23			5
24			11
25			17
26		ReliefFAtributeEval	1
27			3
28			5
29			11
30			17
31	Año completo	Ninguno	1
32			3
33			5
34			11
35			17
36		CfsSubsetEval	1
37			3
38			5
39			11
40			17

41		ReliefFAtributeEval	1
42			3
43			5
44			11
45			17

### Anexo 5. Variantes del NARX experimentadas en MATLAB

Variante	Temporada del dataset	Filtro de selección de atributos	Semilla de generación aleatoria
1	Temporada alta	Ninguno	1
2			3
3			5
4			11
5			17
6		CfsSubsetEval	1
7			3
8			5
9			11
10			17
11		ReliefFAtributeEval	1
12			3
13			5
14			11
15			17
16	Temporada baja	Ninguno	1
17			3
18			5
19			11
20			17
21		CfsSubsetEval	1
22			3
23			5
24			11
25			17
26		ReliefFAtributeEval	1
27			3
28			5
29			11
30			17
31	Año completo	Ninguno	1
32			3

33			5
34			11
35			17
36		CfsSubsetEval	1
37			3
38			5
39			11
40			17
41		ReliefFAtributeEval	1
42			3
43			5
44			11
45			17

### Anexo 6. Variantes del DT experimentadas en MATLAB

Variante	Temporada del dataset	Filtro de selección de atributos	Semilla de generación aleatoria
1	Temporada alta	Ninguno	1
2			3
3			5
4			11
5			17
6		CfsSubsetEval	1
7			3
8			5
9			11
10			17
11		ReliefFAtributeEval	1
12			3
13			5
14			11
15			17
16	Temporada baja	Ninguno	1
17			3
18			5
19			11
20			17
21		CfsSubsetEval	1
22			3
23			5
24			11

25			17
26		ReliefFAtributeEval	1
27			3
28			5
29			11
30			17
31	Año completo	Ninguno	1
32			3
33			5
34			11
35			17
36			CfsSubsetEval
37			3
38			5
39			11
40			17
41		ReliefFAtributeEval	1
42			3
43			5
44			11
45			17

### Anexo 7. Variantes del SVM experimentadas en MATLAB

Variante	Temporada del dataset	Filtro de selección de atributos	Semilla de generación aleatoria
1	Temporada alta	Ninguno	1
2			3
3			5
4			11
5			17
6			CfsSubsetEval
7			3
8			5
9			11
10			17
11		ReliefFAtributeEval	1
12			3
13			5
14			11
15			17
16	Temporada baja		Ninguno

17			3
18			5
19			11
20			17
21		CfsSubsetEval	1
22			3
23			5
24			11
25			17
26		ReliefFAtributeEval	1
27			3
28			5
29			11
30			17
31	Año completo	Ninguno	1
32			3
33			5
34			11
35			17
36		CfsSubsetEval	1
37			3
38			5
39			11
40			17
41		ReliefFAtributeEval	1
42			3
43			5
44			11
45			17

**Anexo 8.** Resultados de las variantes del modelo MLP entrenado con el dataset de temporada alta sin selección de atributos.

Variante	Algoritmo de entrenamiento	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
1	entrenamiento de retropropagación BFGS quasi-Newton	1	0.098367	0.10063
2		3	0.093066	0.093539
3		5	0.09823	0.10328
4		11	0.10277	0.10707
5		17	0.10735	0.1082
6		1	0.10729	0.11224

7	entrenamiento de retropropagación con regularización bayesiana	3	0.10462	0.10839
8		5	0.10466	0.10771
9		11	0.10464	0.1076
10		17	0.10471	0.10973
11	entrenamiento de retropropagación del gradiente conjugado con reinicio Powell-Beale	1	0.097953	0.10188
12		3	0.10105	0.10618
13		5	0.099678	0.10363
14		11	0.10408	0.10166
15		17	0.098272	0.10251
16	entrenamiento de retropropagación del gradiente conjugado con actualización Fletcher-Reeves	1	0.10136	0.10433
17		3	0.11056	0.11066
18		5	0.09652	0.09692
19		11	0.10615	0.10464
20		17	0.10405	0.10624
21	entrenamiento de retropropagación del gradiente descendente	1	0.14715	0.14212
22		3	0.15356	0.15568
23		5	0.14134	0.1438
24		11	0.15796	0.15527
25		17	0.14086	0.14048

**Anexo 9.** Resultados de las variantes del modelo FFNN entrenado con el dataset de temporada alta sin selección de atributos.

Variante	Arquitectura	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
1	FFNN	1	0.091255	0.092846
2		3	0.093566	0.099511
3		5	0.083821	0.082301
4		11	0.090372	0.094508
5		17	0.088494	0.093462
6	FFNN en cascada	1	0.087055	0.085131
7		3	0.094286	0.09623
8		5	0.079252	0.09182
9		11	0.093628	0.092426
10		17	0.091396	0.093234

**Anexo 10.** Resultados de las variantes del modelo MLP entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Algoritmo de entrenamiento	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
26	entrenamiento de retropropagación BFGS quasi-Newton	1	0.10285	0.10222
27		3	0.099347	0.10054
28		5	0.10701	0.10644
29		11	0.10523	0.10385
30		17	0.10513	0.10736
31	entrenamiento de retropropagación con regularización bayesiana	1	0.096354	0.092448
32		3	0.1093	0.10429
33		5	0.10301	0.10919
34		11	0.1027	0.10542
35		17	0.09624	0.098735
36	entrenamiento de retropropagación del gradiente conjugado con reinicio Powell-Beale	1	0.099373	0.10148
37		3	0.099419	0.10032
38		5	0.10539	0.10689
39		11	0.10208	0.10497
40		17	0.10855	0.11446
41	entrenamiento de retropropagación del gradiente conjugado con actualización Fletcher-Reeves	1	0.098545	0.099697
42		3	0.10281	0.10438
43		5	0.10826	0.10639
44		11	0.10413	0.10169
45		17	0.10265	0.10576
46	entrenamiento de retropropagación del gradiente descendente	1	0.16287	0.15707
47		3	0.13831	0.13545
48		5	0.14635	0.14221
49		11	0.14142	0.13853
50		17	0.13974	0.1387

**Anexo 11.** Resultados de las variantes del modelo FFNN entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Arquitectura	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
11	FFNN	1	0.096837	0.096551
12		3	0.10997	0.1145
13		5	0.093443	0.09372
14		11	0.091184	0.0933
15		17	0.10124	0.1024
16	FFNN en cascada	1	0.10579	0.10886
17		3	0.092503	0.094642
18		5	0.09658	0.094632
19		11	0.098842	0.10021
20		17	0.10592	0.1075

**Anexo 12.** Resultados de las variantes del modelo MLP entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka ReliefFAttributeEval.

Variante	Algoritmo de entrenamiento	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
51	entrenamiento de retropropagación BFGS quasi-Newton	1	0.10543	0.11085
52		3	0.099079	0.096121
53		5	0.096499	0.10242
54		11	0.096805	0.096312
55		17	0.10426	0.1075
56	entrenamiento de retropropagación con regularización bayesiana	1	0.10462	0.10912
57		3	0.093923	0.093031
58		5	0.10466	0.10767
59		11	0.10464	0.1076
60		17	0.10471	0.10973
61	entrenamiento de retropropagación del gradiente conjugado con reinicio Powell-Beale	1	0.10399	0.10595
62		3	0.098133	0.094923
63		5	0.096275	0.099965
64		11	0.096888	0.098148
65		17	0.098824	0.10335
66	entrenamiento de retropropagación del gradiente	1	0.10022	0.10083
67		3	0.099513	0.096519
68		5	0.10142	0.1042

<b>69</b>	conjugado con actualización Fletcher-Reeves	11	0.09734	0.097287
<b>70</b>		17	0.094858	0.10084
<b>71</b>	entrenamiento de retropropagación del gradiente descendente	1	0.1655	0.16442
<b>72</b>		3	0.13951	0.13957
<b>73</b>		5	0.14864	0.14756
<b>74</b>		11	0.15255	0.14454
<b>75</b>		17	0.16397	0.17023

**Anexo 13.** Resultados de las variantes del modelo FFNN entrenado con el dataset de temporada alta con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval.

<b>Variante</b>	Arquitectura	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>21</b>	FFNN	1	0.08954	0.08929
<b>22</b>		3	0.097213	0.095671
<b>23</b>		5	0.090403	0.097204
<b>24</b>		11	0.085788	0.083756
<b>25</b>		17	0.090565	0.094699
<b>26</b>	FFNN en cascada	1	0.085151	0.080983
<b>27</b>		3	0.095098	0.095725
<b>28</b>		5	0.12449	0.11329
<b>29</b>		11	0.088424	0.085238
<b>30</b>		17	0.092405	0.09244

**Anexo 14.** Resultados de las variantes del modelo MLP entrenado con el dataset de temporada baja sin selección de atributos.

<b>Variante</b>	Algoritmo de entrenamiento	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>76</b>	entrenamiento de retropropagación BFGS quasi-Newton	1	0.10729	0.10452
<b>77</b>		3	0.10636	0.10636
<b>78</b>		5	0.11518	0.115
<b>79</b>		11	0.11945	0.12015
<b>80</b>		17	0.11133	0.11302
<b>81</b>		1	0.073976	0.16539

<b>82</b>	entrenamiento de retropropagación con regularización bayesiana	3	0.069124	0.15815
<b>83</b>		5	0.068394	0.15113
<b>84</b>		11	0.07819	0.14523
<b>85</b>		17	0.081577	0.13033
<b>86</b>	entrenamiento de retropropagación del gradiente conjugado con reinicio Powell-Beale	1	0.10765	0.10584
<b>87</b>		3	0.10688	0.10763
<b>88</b>		5	0.12137	0.12111
<b>89</b>		11	0.11754	0.11741
<b>90</b>		17	0.11004	0.11014
<b>91</b>	entrenamiento de retropropagación del gradiente conjugado con actualización Fletcher-Reeves	1	0.10744	0.10516
<b>92</b>		3	0.10701	0.10779
<b>93</b>		5	0.11206	0.1171
<b>94</b>		11	0.11732	0.11542
<b>95</b>		17	0.11093	0.11044
<b>96</b>	entrenamiento de retropropagación del gradiente descendente	1	0.16418	0.17069
<b>97</b>		3	0.17077	0.17009
<b>98</b>		5	0.1714	0.1702
<b>99</b>		11	0.16031	0.15559
<b>100</b>		17	0.1501	0.15277

**Anexo 15.** Resultados de las variantes del modelo FFNN entrenado con el dataset de temporada baja sin selección de atributos.

<b>Variante</b>	Arquitectura	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>31</b>	FFNN	1	0.089586	0.089823
<b>32</b>		3	0.10601	0.10369
<b>33</b>		5	0.091484	0.095193
<b>34</b>		11	0.10243	0.10103
<b>35</b>		17	0.10042	0.10097
<b>36</b>	FFNN en cascada	1	0.099585	0.099281
<b>37</b>		3	0.091065	0.089485
<b>38</b>		5	0.090887	0.098527
<b>39</b>		11	0.10125	0.10504
<b>40</b>		17	0.074651	0.07626

**Anexo 16.** Resultados de las variantes del modelo MLP entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Algoritmo de entrenamiento	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
101	entrenamiento de retropropagación BFGS quasi-Newton	1	0.11575	0.1152
102		3	0.10942	0.10782
103		5	0.11321	0.11225
104		11	0.11122	0.11012
105		17	0.10852	0.10771
106	entrenamiento de retropropagación con regularización bayesiana	1	0.070716	0.10406
107		3	0.082722	0.11487
108		5	0.072287	0.1009
109		11	0.087602	0.10906
110		17	0.07611	0.112
111	entrenamiento de retropropagación del gradiente conjugado con reinicio Powell-Beale	1	0.11569	0.11494
112		3	0.10876	0.10792
113		5	0.10895	0.1087
114		11	0.10917	0.10786
115		17	0.10218	0.10205
116	entrenamiento de retropropagación del gradiente conjugado con actualización Fletcher-Reeves	1	0.10127	0.10396
117		3	0.10822	0.10902
118		5	0.10903	0.10845
119		11	0.10706	0.10204
120		17	0.10497	0.10519
121	entrenamiento de retropropagación del gradiente descendente	1	0.1679	0.16637
122		3	0.14745	0.14953
123		5	0.15328	0.16219
124		11	0.15253	0.15173
125		17	0.14811	0.14876

**Anexo 17.** Resultados de las variantes del modelo FFNN entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Arquitectura	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
41	FFNN	1	0.080613	0.08683
42		3	0.090433	0.089846
43		5	0.077559	0.078644
44		11	0.089179	0.099585
45		17	0.090354	0.091771
46	FFNN en cascada	1	0.086284	0.088035
47		3	0.098218	0.10089
48		5	0.083226	0.08098
49		11	0.092885	0.094526
50		17	0.098147	0.098686

**Anexo 18.** Resultados de las variantes del modelo MLP entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka ReliefFAttributeEval.

Variante	Algoritmo de entrenamiento	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
126	entrenamiento de retropropagación BFGS quasi-Newton	1	0.11639	0.11469
127		3	0.11093	0.11154
128		5	0.10668	0.10772
129		11	0.11514	0.11812
130		17	0.11098	0.11149
131	entrenamiento de retropropagación con regularización bayesiana	1	0.070069	0.14908
132		3	0.067977	0.16446
133		5	0.066869	0.13777
134		11	0.079489	0.1463
135		17	0.083203	0.13779
136	entrenamiento de retropropagación del gradiente conjugado con reinicio Powell-Beale	1	0.10401	0.10592
137		3	0.10834	0.10997
138		5	0.10619	0.10764
139		11	0.11408	0.11976
140		17	0.11182	0.11278
141	entrenamiento de retropropagación del gradiente	1	0.11754	0.11504
142		3	0.10892	0.10727
143		5	0.10779	0.10902

<b>144</b>	conjugado con actualización Fletcher-Reeves	11	0.1142	0.12056
<b>145</b>		17	0.11081	0.11361
<b>146</b>	entrenamiento de retropropagación del gradiente descendente	1	0.15444	0.16044
<b>147</b>		3	0.16601	0.16489
<b>148</b>		5	0.14351	0.14564
<b>149</b>		11	0.14771	0.14891
<b>150</b>		17	0.14995	0.15588

**Anexo 19.** Resultados de las variantes del modelo FFNN entrenado con el dataset de temporada baja con la selección de atributos mediante el evaluador de Weka ReliefFAttributeEval.

<b>Variante</b>	Arquitectura	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>51</b>	FFNN	1	0.095364	0.093913
<b>52</b>		3	0.10972	0.10784
<b>53</b>		5	0.10379	0.10278
<b>54</b>		11	0.090083	0.089656
<b>55</b>		17	0.089109	0.09345
<b>56</b>	FFNN en cascada	1	0.098889	0.099198
<b>57</b>		3	0.092674	0.098047
<b>58</b>		5	0.087758	0.089404
<b>59</b>		11	0.10115	0.099457
<b>60</b>		17	0.094093	0.094851

**Anexo 20.** Resultados de las variantes del modelo MLP entrenado con el dataset del año completo sin selección de atributos.

<b>Variante</b>	Algoritmo de entrenamiento	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>151</b>	entrenamiento de retropropagación BFGS quasi-Newton	1	0.10402	0.10456
<b>152</b>		3	0.10231	0.10123
<b>153</b>		5	0.09537	0.096118
<b>154</b>		11	0.096493	0.095436
<b>155</b>		17	0.10419	0.10888
<b>156</b>		1	0.069761	0.11124

<b>157</b>	entrenamiento de retropropagación con regularización bayesiana	3	0.075787	0.10567
<b>158</b>		5	0.068824	0.11082
<b>159</b>		11	0.068156	0.11828
<b>160</b>		17	0.074481	0.10955
<b>161</b>	entrenamiento de retropropagación del gradiente conjugado con reinicio Powell-Beale	1	0.1096	0.11062
<b>162</b>		3	0.10357	0.10273
<b>163</b>		5	0.095634	0.095189
<b>164</b>		11	0.093951	0.092786
<b>165</b>		17	0.096987	0.098126
<b>166</b>	entrenamiento de retropropagación del gradiente conjugado con actualización Fletcher-Reeves	1	0.10331	0.10592
<b>167</b>		3	0.10521	0.10437
<b>168</b>		5	0.094741	0.09536
<b>169</b>		11	0.0967	0.095801
<b>170</b>		17	0.10146	0.10223
<b>171</b>	entrenamiento de retropropagación del gradiente descendente	1	0.14866	0.15027
<b>172</b>		3	0.15054	0.14749
<b>173</b>		5	0.1451	0.14553
<b>174</b>		11	0.15299	0.1537
<b>175</b>		17	0.18393	0.17429

**Anexo 21.** Resultados de las variantes del modelo FFNN entrenado con el dataset del año completo sin selección de atributos.

<b>Variante</b>	Arquitectura	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>61</b>	FFNN	1	0.09281	0.09413
<b>62</b>		3	0.094241	0.092866
<b>63</b>		5	0.092143	0.091446
<b>64</b>		11	0.090881	0.089426
<b>65</b>		17	0.09393	0.095828
<b>66</b>	FFNN en cascada	1	0.087442	0.086437
<b>67</b>		3	0.092196	0.093119
<b>68</b>		5	0.098191	0.098289
<b>69</b>		11	0.094086	0.093245
<b>70</b>		17	0.090765	0.094634

**Anexo 22.** Resultados de las variantes del modelo MLP entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Algoritmo de entrenamiento	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
176	entrenamiento de retropropagación BFGS quasi-Newton	1	0.11438	0.11361
177		3	0.11385	0.11301
178		5	0.11599	0.11523
179		11	0.1155	0.11558
180		17	0.11492	0.11354
181	entrenamiento de retropropagación con regularización bayesiana	1	0.11356	0.11401
182		3	0.11384	0.1148
183		5	0.11368	0.11569
184		11	0.11364	0.11537
185		17	0.11396	0.11446
186	entrenamiento de retropropagación del gradiente conjugado con reinicio Powell-Beale	1	0.11471	0.11409
187		3	0.11385	0.11337
188		5	0.1151	0.11451
189		11	0.11428	0.11452
190		17	0.11472	0.11354
191	entrenamiento de retropropagación del gradiente conjugado con actualización Fletcher-Reeves	1	0.11477	0.11361
192		3	0.11408	0.11359
193		5	0.11515	0.1145
194		11	0.11484	0.11516
195		17	0.11496	0.11388
196	entrenamiento de retropropagación del gradiente descendente	1	0.15881	0.15334
197		3	0.1322	0.13026
198		5	0.12657	0.12534
199		11	0.16734	0.16675
200		17	0.14501	0.14742

**Anexo 23.** Resultados de las variantes del modelo FFNN entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka CfsSubsetEval.

Variante	Arquitectura	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
71	FFNN	1	0.11174	0.11215
72		3	0.11229	0.11172
73		5	0.11398	0.1132
74		11	0.10968	0.10876
75		17	0.11455	0.11358
76	FFNN en cascada	1	0.11256	0.11188
77		3	0.1135	0.11334
78		5	0.11443	0.11358
79		11	0.11247	0.11208
80		17	0.11414	0.11267

**Anexo 24.** Resultados de las variantes del modelo MLP entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka ReliefFAttributeEval.

Variante	Algoritmo de entrenamiento	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
201	entrenamiento de retropropagación BFGS quasi-Newton	1	0.10402	0.10456
202		3	0.10231	0.10123
203		5	0.09537	0.096118
204		11	0.096493	0.095436
205		17	0.10419	0.10888
206	entrenamiento de retropropagación con regularización bayesiana	1	0.069761	0.11124
207		3	0.075787	0.10567
208		5	0.068824	0.11082
209		11	0.068156	0.11828
210		17	0.074481	0.10955
211	entrenamiento de retropropagación del gradiente conjugado con reinicio Powell-Beale	1	0.1096	0.11062
212		3	0.10357	0.10273
213		5	0.095634	0.095189
214		11	0.093951	0.092786
215		17	0.096987	0.098126
216	entrenamiento de retropropagación del gradiente	1	0.10331	0.10592
217		3	0.10521	0.10437
218		5	0.094741	0.09536

<b>219</b>	conjugado con actualización Fletcher-Reeves	11	0.0967	0.095801
<b>220</b>		17	0.10146	0.10223
<b>221</b>	entrenamiento de retropropagación del gradiente descendente	1	0.14866	0.15027
<b>222</b>		3	0.15054	0.14749
<b>223</b>		5	0.1451	0.14553
<b>224</b>		11	0.15299	0.1537
<b>225</b>		17	0.18393	0.17429

**Anexo 25.** Resultados de las variantes del modelo FFNN entrenado con el dataset del año completo con la selección de atributos mediante el evaluador de Weka ReliefFAtributeEval.

<b>Variante</b>	Arquitectura	Semilla de generación aleatoria	RMSE de la etapa de entrenamiento	RMSE de la etapa de prueba
<b>81</b>	FFNN	1	0.09281	0.09413
<b>82</b>		3	0.094241	0.092866
<b>83</b>		5	0.092143	0.091446
<b>84</b>		11	0.090881	0.089426
<b>85</b>		17	0.09393	0.095828
<b>86</b>	FFNN en cascada	1	0.087442	0.086437
<b>87</b>		3	0.092196	0.093119
<b>88</b>		5	0.098191	0.098289
<b>89</b>		11	0.094086	0.093245
<b>90</b>		17	0.090765	0.094634

## Anexo 26. Código de la implementación en MATLAB del modelo propuesto para la temporada alta

```

function [] = tree()

M = csvread("TA - estandarizado.csv"); %nombre del csv.
X = M(:,1:size(M,2)-1); %valores de las variables predictoras%
y = M(:,size(M,2)); %valores de la variable objetivo%
rng(3) %semilla para la generacion aleatoria

cvp = cvpartition(size(M,1), 'KFold',10); %Create cross-validation
partition for data
rmse = zeros(cvp.NumTestSets,1); %Create array of all zeros
rmseTrain = zeros(cvp.NumTestSets,1); %Create array of all zeros

rand1 = zeros(cvp.NumTestSets*5,1);
rand2 = zeros(cvp.NumTestSets*5,1);

for i = 1:cvp.NumTestSets
    trIdx = cvp.training(i);
    teIdx = cvp.test(i);
    %Xs y Ys de entrenamiento de esta iteración del cross-validation%
    Xi_train = X(trIdx,:)';
    yi_train = y(trIdx,:)';
    tree = fitrtree(Xi_train',yi_train');%creo el árbol de decisión
    %Xs y Ys de evaluación de esta iteración del cross-validation%
    Xi_test = X(teIdx,:)';
    yi_test = y(teIdx,:)';
    obtenidosTrain = predict(tree, Xi_train');
    obtenidos = predict(tree, Xi_test');
    si=length(Xi_test);
    rand=randperm(si,5);%generar un arreglo de 5 números aleatorios entre
1 y length(TsNew)
    i=i-1;
    for j = 1:5
        rand1((i*5)+j)=yi_test(rand(j));
        rand2((i*5)+j)=obtenidos(rand(j));
    end
    i=i+1;
    rmseTrain(i) = getRMSE(obtenidosTrain,yi_train); %aquí guardas el rmse
de esta iteracion%
    rmse(i) = getRMSE(obtenidos,yi_test); %aquí guardas el rmse de esta
iteracion%
end

```

```

rmsef = mean(rmse);
rmsefTrain = mean(rmseTrain);

plot(rand1, 'r');
    hold on;
    plot(rand2, 'b');
    title('Modelo Árbol de Decisión');
    legend('Salida Deseada', 'Salida Obtenida');

save('pretrained_model_alta.mat', 'tree');

end

```

### Anexo 27. Código de la implementación en MATLAB del modelo propuesto para la temporada baja

```

function [] = tree()

M = csvread("TB - estandarizado.csv"); %nombre del csv.
X = M(:,1:size(M,2)-1); %valores de las variables predictoras%
y = M(:,size(M,2)); %valores de la variable objetivo%
rng(1) %semilla para la generacion aleatoria

cvp = cvpartition(size(M,1), 'KFold', 10); %Create cross-validation
partition for data
rmse = zeros(cvp.NumTestSets,1); %Create array of all zeros
rmseTrain = zeros(cvp.NumTestSets,1); %Create array of all zeros

rand1 = zeros(cvp.NumTestSets*5,1);
rand2 = zeros(cvp.NumTestSets*5,1);

for i = 1:cvp.NumTestSets
    trIdx = cvp.training(i);
    teIdx = cvp.test(i);
    %Xs y Ys de entrenamiento de esta iteración del cross-validation%
    Xi_train = X(trIdx,:)';
    yi_train = y(trIdx,:)';
    tree = fitrtree(Xi_train', yi_train'); %creo el árbol de decisión
    %Xs y Ys de evaluación de esta iteración del cross-validation%
    Xi_test = X(teIdx,:)';
    yi_test = y(teIdx,:)';
    obtenidosTrain = predict(tree, Xi_train');
    obtenidos = predict(tree, Xi_test');
    si=length(Xi_test);
    rand=randperm(si,5); %generar un arreglo de 5 números aleatorios entre
1 y length(TsNew)

```

```
i=i-1;
    for j = 1:5
        rand1((i*5)+j)=yi_test(rand(j));
        rand2((i*5)+j)=obtenidos(rand(j));
    end
    i=i+1;
    rmseTrain(i) = getRMSE(obtenidosTrain,yi_train); %aqui guardas el
rmse de esta iteracion%
    rmse(i) = getRMSE(obtenidos,yi_test); %aqui guardas el rmse de esta
iteracion%
end

rmsef = mean(rmse);
rmsefTrain = mean(rmseTrain);

plot(rand1,'r');
    hold on;
    plot(rand2,'b');
    title('Modelo Árbol de Decisión');
    legend('Salida Deseada','Salida Obtenida');

save('pretrained_model_baja.mat', 'tree');

end
```