Dimensionamiento de Microrredes Aisladas a partir de Técnicas de Regresión de Aprendizaje Automático

Carlos Eduardo Méndez Mateus

Daniel Santiago Barrera Ariza

Trabajo de grado para optar por el título de Ingeniero Electricista

Director

Dr. Juan Manuel Rey López

Codirector

Dr. Carlos Augusto Fajardo Ariza

Universidad Industrial de Santander

Facultad de Ingenierías Físico-Mecánicas

Escuela de Ingenierías Eléctrica, Electrónica y de Telecomunicaciones

Bucaramanga

2023

Dedicatoria

A mi madre por su apoyo incondicional, motivación y los buenos consejos brindados durante todo el proceso. A mi padre por la confianza, el apoyo emocional e incondicional depositado. Hoy y siempre puedo decir que me han ayudaron a lograr lo que jamás pensé que lograría.

Daniel Santiago Barrera Ariza

Principalmente a mis padres, que me acompañaron en este proceso con total confianza, dedicación y amor. A mis hermanas, que me apoyaron con su consejo, apoyo emocional y finalmente a mi hermano que desde el cielo me ilumina y me guía en cada paso.

Carlos Eduardo Méndez Mateus

Agradecimientos

A Dios, por brindarnos salud y sabiduría para culminar esta etapa tan importante en nuestro desarrollo académico y profesional.

A nuestro director, por la paciencia y dedicación a lo largo de este proyecto. Muchas gracias por compartir sus conocimientos y corregirnos oportunamente.

A nuestro codirector y al ingeniero Iván, por atender nuestras inquietudes.

Tabla de Contenido

Pag.

Lista de Tablas

Pag.

Lista de Figuras

Pag.

Resumen

Título: Dimensionamiento de Microrredes Aisladas a partir de Técnicas de Regresión de Aprendizaje Automático*.

Autores: Carlos Eduardo Méndez Mateus, Daniel Santiago Barrera Ariza**

Palabras claves: Microrredes, aprendizaje automático, energías renovables, precisión, regresión.

Descripción:

En este proyecto de grado se propondrá una metodología alternativa para el dimensionamiento de microrredes aisladas usando técnicas de regresión de aprendizaje automático. Como punto de partida, se obtendrá un set de datos a partir de una formulación clásica de dimensionamiento usando la técnica fuerza bruta para condiciones climáticas y de carga específicas. Luego, se seleccionarán al menos dos estrategias de técnicas de regresión de aprendizaje automático diferentes a redes neuronales, como árboles de decisiones, bosques aleatorios, regresión vectorial, regresión lineal (regresión de cresta o regresión de lazo), o regresión polinómica. Una vez obtenidos los modelos, se propondrá un procedimiento de entrenamiento y evaluación para calcular la relación que existe entre la precisión de los modelos y la reducción de la proporción de datos de entrenamiento.

* Trabajo de grado.

** Facultad de Ingenierías Fisicomecánicas, Escuela E3T. Director Juan Manuel Rey López. Codirector Carlos Augusto Fajardo Ariza

Abstract

Title: Sizing of isolated microgrids using machine learning regression techniques.

Authors: Carlos Eduardo Méndez Mateus, Daniel Santiago Barrera Ariza**

Keywords: Microgrids, Machine Learning, Renewable energies, Precision, Regression

Description:

In this undergraduate project, an alternative methodology for sizing isolated microgrids using machine learning regression techniques will be proposed. As a starting point, a dataset will be obtained from a classical sizing formulation using brute force technique for specific weather and load conditions. Then, at least two different machine learning regression techniques other than neural networks, such as decision trees, random forests, vector regression, linear regression (ridge regression or lasso regression), or polynomial regression, will be selected. Once the models are obtained, a training and evaluation procedure will be proposed to calculate the relationship between model accuracy and the reduction in the proportion of training da

∗ Degree work

∗∗ Faculty of physical and mechanical engineering, E3T school. Director Juan Manuel Rey López. Co-director Carlos Augusto Fajardo Ariza

Introducción

La generación de energía a partir de fuentes renovables presenta un crecimiento acelerado en las últimas décadas. En la actualidad se adelantan proyectos de energías renovables a lo largo del territorio nacional con el fin de garantizar suministro de energía sostenible, este es uno de los retos más importantes a nivel internacional según la IEA (Agencia Internacional de energía) (Vanegas Chamorro, Villicaña Ortíz, and Arrieta Viana 2015).

La matriz energética colombiana se ha caracterizado por ser una de las más limpias del mundo, esto se debe a que aproximadamente el 70% de la generación de energía proviene de fuentes hídricas y 30% de las plantas térmicas fósiles (Ministerio de Minas y Energía and Unidad de Planeación Minero-Energética - UPME 2015). Debido a que Colombia cuenta con baja participación de las energías renovables no convencionales, se encuentra dispuesta a aumentar la participación de estas.

En Colombia, las energías renovables, principalmente la eólica y solar, desempeñarán un papel importante en la transición energética, debido a que favorecen la expansión de un sistema de energía limpio, confiable, resistente y asequible. Para las fuentes de energía renovable, las condiciones geográficas y meteorológicas son de vital importancia, tal es el caso de los sistemas fotovoltaicos. Colombia y países cercanos al ecuador poseen una radiación solar promedio más alta que la mayoría de los países en Europa y Estados Unidos (López et al. 2020).

Según la UPME (Unidad de Planeación Minero-Energética), el país cuenta con regiones como la Guajira que presenta las condiciones más destacadas de todo el territorio Nacional para la instalación de sistemas de generación de energía a partir de la radiación solar (Carvajal-Romo et al. 2019). Además, Colombia, por su ubicación en el Trópico de Cáncer y el Trópico de Capricornio, está sujeta a gran cantidad de vientos en el hemisferio norte y parte del hemisferio sur, por ende, estas zonas poseen gran potencial para la instalación de sistemas eólicos (The Renewables Consulting Group 2022).

La adopción masiva de las energías renovables plantea nuevos retos tecnológicos, debido a que su dependencia climatológica la hace difícilmente controlable. La integración de los sistemas de generación renovables a la red, especialmente la energía eólica y fotovoltaica, comienza a ocasionar impactos en la misma como pueden ser la variación de la tensión del suministro,

presencia de armónicos e incremento en el desbalance entre la potencia activa y reactiva (Bordons, García-Torres, and Valverde 2015). Los sistemas de almacenamiento de energía aparecen como solución tecnológica a los problemas de fluctuaciones de la generación renovable y la aleatoriedad del comportamiento de los consumidores, por consiguiente, se opta por una alternativa llamada microrred (en inglés, microgrid).

Una microrred se define como un pequeño sistema de potencia compuesto principalmente por cargas controladas y no controladas, unidades de generación distribuida, sistemas de almacenamiento de energía (J. Oviedo, J. Bastidas, and J. Solano 2017). Además, cuenta con sistemas de protección coordinados con sistemas de control, haciendo posible su operación de manera aislada o conectada a un sistema de potencia más grande y dispositivos de electrónica de potencia que tendrán un impacto significativo en el diseño y planificación de la microrred (Sandelic et al. 2022).

Las microrredes aisladas se reconocen como una de las soluciones más adecuadas y rentables para la electrificación en zonas rurales y/o alejadas (Rey et al. 2022). El correcto dimensionamiento de sus sistemas de generación y almacenamiento de energía aseguran un uso eficiente de los recursos energéticos disponibles, disminución de costos en las tecnologías de energía renovable debido a los avances tecnológicos y un alto índice de confiabilidad en la microrred instalada (Boutros et al. 2023).

En el caso de la operación aislada, un inconveniente de las microrredes es que, al basarse principalmente en la generación mediante energías renovables, la confiabilidad del sistema puede verse gravemente afectada por las condiciones climáticas de la zona de implementación. Como alternativa, se recomienda considerar la implementación de sistemas híbridos que integren fuentes renovables con múltiples sistemas de almacenamiento y generadores a base de combustibles como sistema de respaldo (Rey et al. 2022). Una de las configuraciones más comunes y confiables en una microrred aislada es aquella conformada por paneles solares, sistemas eólicos, baterías y generadores diésel (Iván Jiménez, juan M. Rey, and German Osma-Pinto 2020).

Las microrredes al estar aisladas de los sistemas de potencia presentan grandes desafíos, uno de ellos es la estabilidad, estos se estudian dependiendo el tipo de microrred, los tipos de unidades de generación distribuida instalados, los parámetros de red, entre otros aspectos (J. Oviedo, J. Bastidas, and J. Solano 2017).

1. Objetivos

1.1. Objetivo general

Diseñar un modelo para el dimensionamiento de una microrred aislada para condiciones climáticas y de carga específicas a partir de dos técnicas de regresión de aprendizaje automático.

1.2. Objetivos específicos

- Implementar una formulación clásica de dimensionamiento usando la técnica fuerza bruta para obtener el set de datos de entrenamiento.
- Entrenar al menos dos modelos usando técnicas de regresión de aprendizaje automático con la información de entrada y salida del set de datos.
- Evaluar el desempeño de los modelos entrenados respecto la formulación clásica.
- Proponer un procedimiento para evaluar la relación que existe entre la precisión de los modelos y la reducción de la proporción de los datos usados para el entrenamiento

2. Marco Teórico

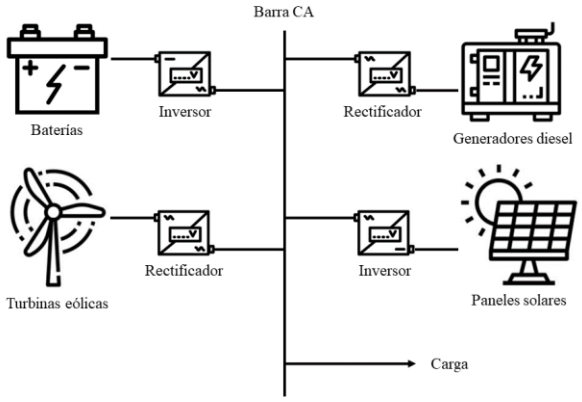
2.1. Dimensionamiento de microrredes

Las unidades de generación de una microrred aislada deben ser capaces de atender la carga instalada con un sistema de respaldo de almacenamiento de energía, con el fin de garantizar fiabilidad y continuidad en el suministro (J.D. Garzón-Hidalgo and A.J. Saavedra-Montes 2017). La forma de generación de energía dependerá de la disponibilidad de recursos de la zona, priorizando la obtención de esta a base de recursos renovables.

Para realizar el dimensionamiento de una microrred, se debe disponer de información previa adquirida de entidades que registren el comportamiento de variables en un periodo de tiempo (Balestrieri, Kahrobaee, and Kim 2021). Los aerogeneradores y paneles solares dependen de las condiciones ambientales del lugar (Khan, Pal, and Saeed 2018), por ende, en Colombia la entidad pertinente la cual proporciona la información relacionada con la radiación solar y la intensidad del viento es el IDEAM (Instituto de Hidrología, Meteorología y Estudios Ambientales).

Adicionalmente se realiza un estudio previo de las condiciones de la carga: Número de usuarios, ubicación, distancia entre usuarios, área promedio de vivienda, área disponible para la instalación de la microrred.

Figura 1.
Elementos de una microrred típica



El dimensionamiento de microrredes tiene como objetivo realizar la selección de los conjuntos de generación y almacenamiento de energía que componen la microrred. Debido a la complejidad y no linealidad del problema de dimensionamiento, se requieren utilizar estrategias que permitan solucionar el problema, con el fin de encontrar una configuración de acuerdo con los criterios de selección (Rey et al. 2019). Adicionalmente, el dimensionamiento de microrredes debe ser confiable en cuanto a la continuidad del suministro, respetuosa con el medio ambiente y demandar un costo lo más bajo sea posible (Kamal, Ashraf, and Fernandez 2023).

2.1.1. Análisis de confiabilidad

Para medir el desempeño de una microrred hibrida en cuanto a confiabilidad, es importante seleccionar criterios de diseño (Dali et al. 2018). Los indicadores de confiabilidad actúan como un filtro para asegurar una configuración adecuada, los más utilizados son el LPSP y LOLH que serán explicados a continuación:

2.1.2. LPSP (Loss of power supply probability)

La probabilidad de pérdida del suministro eléctrico LPSP es un indicador que mide la probabilidad de desabastecimiento de la carga al exceder la capacidad de generación. Se estima hallando la relación de LPS (Perdida de suministro por hora donde la carga no puede ser atendida en su totalidad por generación y almacenamiento) y la demanda por hora (P_{LOAD}) en un tiempo

establecido (Rey et al. 2019). Este indicador es el criterio más utilizado en el dimensionamiento de microrredes hibridas y se calcula a través de la siguiente expresión:

$$LPSP = \frac{\sum LPS}{\sum \text{PLOAD}} x100\% \quad (1)$$

2.1.3. LOLH (Loss of load hour)

Las pérdidas de hora de carga LOLH, cuantifican el porcentaje de horas en que la demanda supera la capacidad de generación (Rey et al. 2022). Este indicador se calcula a través de la siguiente expresión:

$$LOLH = \frac{\sum HLPS}{N} x100\% \quad (2)$$

Donde *HLPS* es el número de horas que la carga no puede ser atendida y *N*, el total de horas estimado en el análisis.

2.1.4. Análisis de costo

El análisis de costo es primordial en el dimensionamiento de microrredes. No solo ayuda a determinar el costo del proyecto, su operación y mantenimiento sino también su viabilidad. Se lleva a cabo una estimación de estos, consiguiendo una eficiente asignación y control de los recursos, evitando costos innecesarios.

El análisis de costo consta de dos componentes:

CAPEX: Son gastos e inversiones asociados con bienes físicos adquiridos. Incluye costos de instalación y reemplazo y salvamento (Rey et al. 2022).

OPEX: El gasto operativo se relaciona con el costo de las operaciones, mantenimientos y servicios.

2.1.5. Costo actual neto

Para realizar el cálculo del costo, se establece una función que defina el valor presente de todos los componentes e inversiones futuras a valor de la moneda actual, teniendo presente las tasas de descuento y la inflación prevista. El costo actual neto (NPC) se utiliza para conocer la rentabilidad de un proyecto, así como su viabilidad económica (Rey et al. 2019). Se calcula a través de la siguiente expresión.

$$NPC = I_0 + \sum Co\,(\frac{1+inf}{1+d})^n \tag{3}$$

Donde *NPC* representa el valor actual neto de la inversión, I_0 es la inversión inicial que incluye la compra e instalación de los equipos al inicio del proyecto, *Co* es el costo del componente que será comprado en *n* años. Adicionalmente, cubre los costos de operación, mantenimiento y combustible necesarios cada año, *inf* el factor de inflación, *d* es la tasa de descuento y *n* el número de años de análisis del proyecto.

La inversión inicial I_0 es el producto de la cantidad de unidades de generación y almacenamiento de energía, multiplicado por su respectivo precio de compra e instalación (Iván Edgardo Jiménez Vargas 2021). Se calcula con la siguiente expresión:

$$I_0 = P_{gd}N_d + P_wN_w + P_PN_P + P_bN_b \tag{4}$$

Donde P_{gd}, P_w, P_p y P_b, son el precio de compra e instalación de un generador diesel, una turbina eólica, un panel solar y una batería, respectivamente, mientras que N_d, N_w, N_p y N_b representan el número de generadores diesel, turbinas eólicas, paneles solares y baterías que componen la microrred.

El costo de operación *Co*, es la acumulación de los gastos de la microrred tras un año de operación.

$$C_0 = C_{FC} + C_{OM} + C_{RE} \tag{5}$$

Siendo C_{FC} el costo del combustible consumido, C_{OM} el costo de operación y mantenimiento y C_{RE} el costo de reemplazo de los componentes.

$$C_{FC} = P_dFC_{gd} \tag{6}$$

$$C_{OM} = OM_{gd}OH_{gd}N_d + OM_wN_w + OM_PN_P + OM_bN_bL \quad (7)$$

$$C_{RE} = 0{,}8(P_{gd}N_{dRE} + P_wN_{wRE} + P_PN_{PRE} + P_bN_{bRE}) \quad (8)$$

P_d: Precio del diesel

FC_{gd}: Cantidad de combustible consumido

OM_{gd}: Costo de operación y mantenimiento del generador diesel en $/hora

OH_{gd}: Horas de operación del generador diesel

OM_w: Costo de operación y mantenimiento de las turbinas eólicas

OM_P: Costo de operación y mantenimiento de los paneles solares

OM_b: Costo de operación y mantenimiento de las baterías

N_{dRE}: Número de unidades reemplazadas de generadores diesel

N_{wRE}: Número de unidades reemplazadas de turbinas eólicas

N_{PRE}: Número de unidades reemplazadas de panales solares

N_{bRE}: Número de unidades reemplazadas de baterías

2.2. Métodos de regresión

2.2.1. Regresión Lineal

Para el caso de la regresión lineal, se cuenta con dos métodos de estudio:

- Regresión lineal simple
- Regresión lineal múltiple

Por la complejidad del problema, se utilizó el método de regresión lineal múltiple, pero para poder entender esta última, es necesario comprender como funciona una regresión lineal simple.

2.2.1.1. Regresión Lineal Simple.

La regresión lineal simple es un método estadístico que estudia la relación lineal entre dos variables (Andreas C. Muller and Sarah Guido, n.d.), del cual podemos señalar varios puntos:

- La regresión genera una ecuación o modelo el cual permite predecir el valor de una variable a partir de otra gracias a la relación que exista entre dos variables.
- Generalmente, se busca una correlación entre las variables para poder generar un modelo de regresión; este modelado es cambiante con respecto a cuál variable se estima dependiente de la otra.
- Para el método de regresión lineal es común controlar una de las dos variables a relacionar; la variable "*X*" suele ser controlada, mientras que la variable "*Y*" es la medida.

A partir de esto podemos definir que la regresión lineal simple consiste en generar un modelo de regresión, este modelo se representa con una ecuación de una línea recta que trata de explicar y mostrar la relación que existe entre la variable dependiente e independiente (Andreas C. Muller and Sarah Guido, n.d.). La variable independiente se identifica con *X* y la variable dependiente se identifica con *Y*.

Este modelo se puede representar con la siguiente ecuación:

$$Yi = \beta_0 + \beta_1 X_1 + e \tag{9}$$

Donde β_0 es el intercepto, β_1 es la pendiente y e es el error aleatorio, que representa la diferencia entre el valor ajustado por la línea y el valor real, va tomando el efecto de todas las variables que influyentes en *Y*, pero no las que se encuentran incluidas en el modelo como predictores (Douglas C. Montgomery, Elizabeth A. Peck, and G. Geoffrey Vining 2021).

En el mayor de los casos, tanto el valor de β_0 como el valor de β_1 son desconocidos, es por ello por lo que los valores iniciales ($\hat{\beta}_0\, y\, \hat{\beta}_1$) se obtienen a partir una muestra. Estos valores se denominan coeficientes de regresión ya que toman los valores que hacen que la suma de cuadrados residual sea minimizada, obteniendo la línea que pasa más cerca a todos los puntos.

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x \tag{10}$$

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n}(x_i - \bar{x})^2} = \frac{S_y}{S_x} R \tag{11}$$

$$\hat{\beta}_0 = \hat{y} + \hat{\beta}_1 \bar{x} \tag{12}$$

Donde S_y y S_x son las desviaciones de cada variable, *R* es el coeficiente de correlación, y $\hat{\beta}_0$ es el valor esperado de la variable *y* cuando *x* es igual a cero (la *x* es un dato necesario para generar la línea, pero en general su interpretación no es practica ya que en varias situaciones *x* no puede ser cero) (Douglas C. Montgomery, Elizabeth A. Peck, and G. Geoffrey Vining 2021). Una vez comprendido esto, podemos abarcar la regresión lineal múltiple.

2.2.1.2. Regresión Lineal Múltiple.

La regresión lineal múltiple es un modelado lineal en el que el valor de la variable dependiente Y, es determinado por un conjunto de variables dependientes o también llamados predictores (X_1, X_2, X_3, … X_n) (Trevor Hastie, Robert Tibshirani, and Jerome Friedman 2009).

Este modelo se puede representar con la siguiente ecuación.

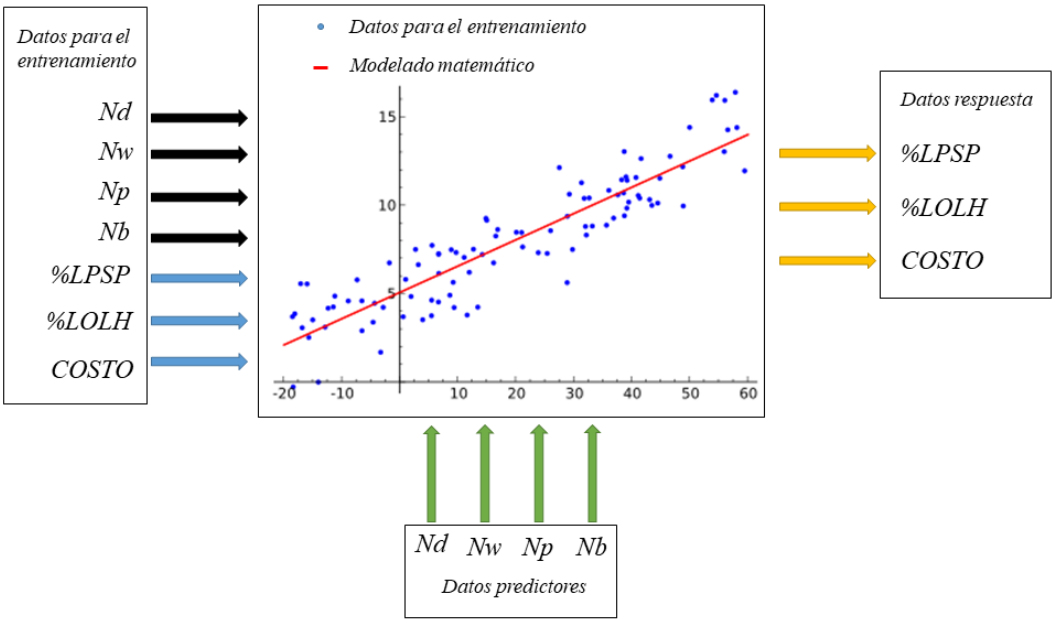
$$Yi = (\beta_0 + \beta_1 X_{1i} + \beta_2 X_{2i} + \cdots + \beta_n X_{ni}) + e_i \quad (13)$$

Donde β_0 es el intercepto en *Y* cuando todas las variables independientes son cero, β_i es el promedio del incremento unitario de la variable independiente X_i sobre la variable dependiente *Y* mientras se mantienen constantes las demás variables (también puede ser llamado coeficiente de regresión parcial) y e_i es la diferencia entre el valor original y el estimado por el modelo.

Es de vital importancia entender que los coeficientes de regresión parcial dependen de las unidades en las que es medida su variable independiente, lo que significa que la magnitud de esta no está vinculada con la importancia de cada variable independiente (Trevor Hastie, Robert Tibshirani, and Jerome Friedman 2009).

En la Figura 4 podemos observar el funcionamiento de la regresión lineal múltiple, donde las flechas de color negro y azul representan los datos de entrada *X* y los datos de salida *Y* usados para el entrenamiento respectivamente. Luego de obtener el modelo entrenado, el cual es representado con una línea roja, se evalúan con los datos predictores mostrados con las flechas de color verde y se obtienen los resultados de la predicción simbolizados con las flechas de color amarillo.

Figura 2.
Ejemplo Regresión lineal en el problema de dimensionamiento de microrredes



2.2.2. Regresión Polinómica

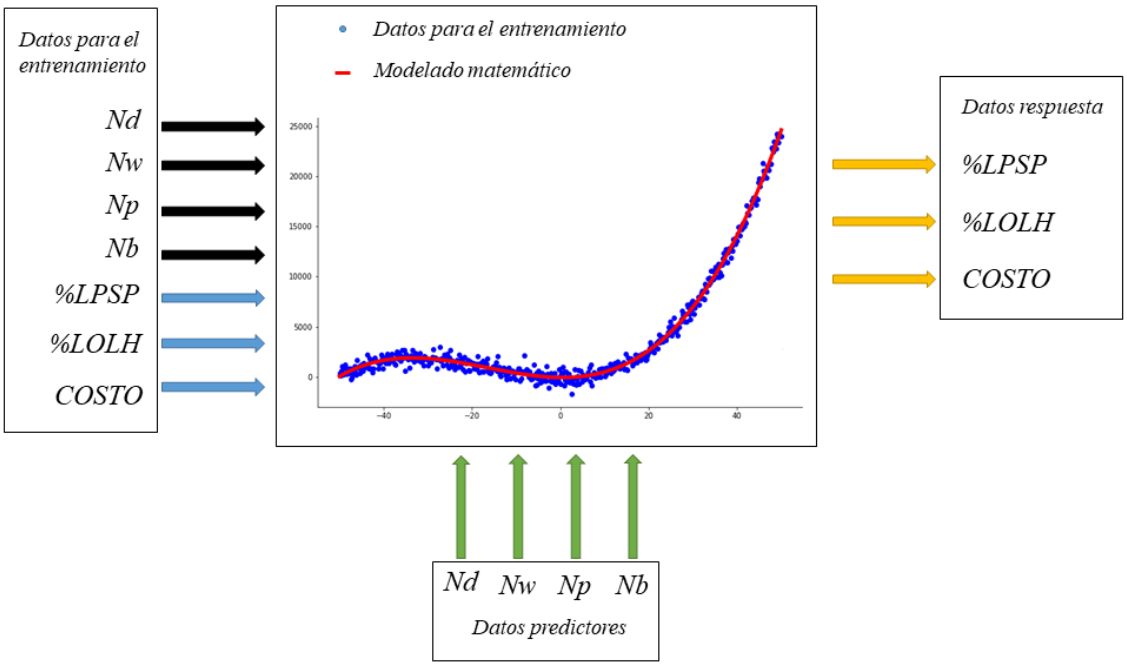
La regresión polinómica parte de un modelo lineal en el que se consigue añadir curvatura, introduciendo nuevas variables independientes que se obtienen al elevar las variables independientes originales a distintas potencias (Sebastian Raschka and Vahid Mirjalili 2017).

$$y = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_j + \beta_4 x_j^2 + \cdots \quad (14)$$

Los modelos polinómicos se ajustan mediante regresión lineal por mínimos cuadrados, esto se debe a que, a pesar de no generar modelos no lineales, el modelo no deja de ser una ecuación lineal con las variables independientes $x^2, x^3, \ldots x^n$ (Sebastian Raschka and Vahid Mirjalili 2017).

En la regresión polinómica mostrada en la Figura 5, observamos una interacción similar a la regresión lineal múltiple; las flechas de color negro y azul representan los datos de entrada *X* y los datos de salida *Y* usados para el entrenamiento respectivamente. Se obtiene el modelo entrenado el cual es trazado con una línea roja, se evalúan con los datos predictores simbolizados con las flechas de color verde, y se obtienen los resultados de la predicción representados con las flechas de color amarillo.

Figura 3.
Ejemplo Regresión polinómica en el problema de dimensionamiento de microrredes



2.2.3. Regresión Con Random Forest

Un modelo de bosque aleatorio o también llamado *Random Forest* consta de una serie de árboles de decisión individuales donde cada árbol selecciona un dato ligeramente diferente a los datos de entrenamiento de arranque. El resultado de la predicción se obtiene sumando las predicciones de todos los árboles individuales que componen el modelo.

Para poder entender el funcionamiento de un bosque aleatorio, primero debemos entender cómo funciona un árbol de decisión.

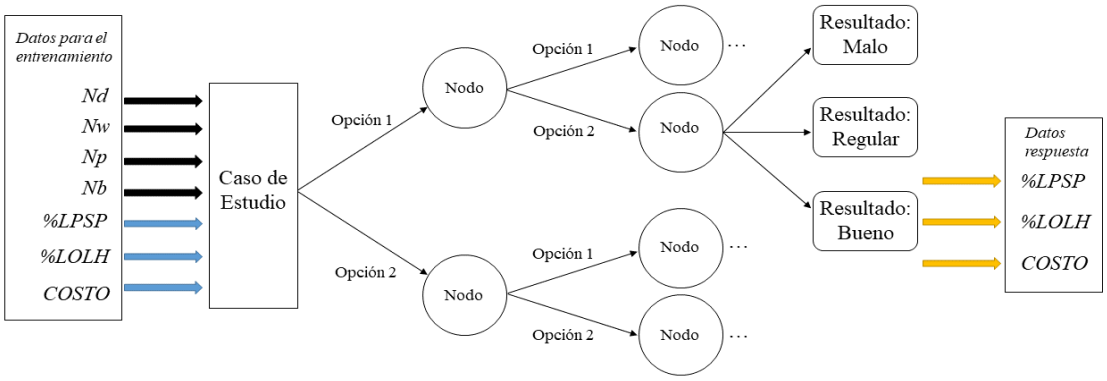
2.2.3.1. Arboles De Decisión.

Los árboles de decisión son modelos predictivos construidos a partir de reglas binarias que distribuyen las observaciones según sus atributos, prediciendo el valor de la respuesta (Trevor Hastie, Robert Tibshirani, and Jerome Friedman 2009). Muchos de los métodos de predicción producen modelos en los que una sola ecuación se aplica a todas las muestras, pero en el caso de usar múltiples predictores que interactuar de manera compleja o no lineal, es difícil encontrar un único modelo global que pueda reflejar la relación entre las variables; debido a esto, los métodos

de M.L. basados en arboles incluyen un conjunto de técnicas supervisadas que logran dividir el espacio de los predictores en regiones simples.

Los árboles de decisión para regresión se aplican cuando la variable es continua, es decir que el entrenamiento va distribuyéndose por nodos, generando una estructura de árbol hasta alcanzar un nodo terminal. La predicción es la media de la variable obtenida de las observaciones del entrenamiento que se encuentren en el mismo nodo terminal.

Figura 4.
Ejemplo Árboles de decisiones en el problema de dimensionamiento de microrredes



El entrenamiento de un árbol de regresión se divide en dos fases:

1. División consecutiva del espacio predictor que produce los nodos terminales. Teóricamente las regiones pueden tener cualquier forma, su limitación a regiones rectangulares simplifica el procedimiento de construcción e interpretación.
2. Predicción de la variable respuesta de cada región.

Es bastante sencillo, pero es necesario establecer la metodología que permita crear las regiones o así mismo, decidir dónde van a ir las divisiones, en que predictores y en que valores de estos.

El criterio más común utilizado para identificar las divisiones es la suma residual de cuadrados o *Residual sum of Squares (RSS).* El objetivo principal es encontrar las regiones que minimizan el RSS (Kevin P. Murphy 2012).

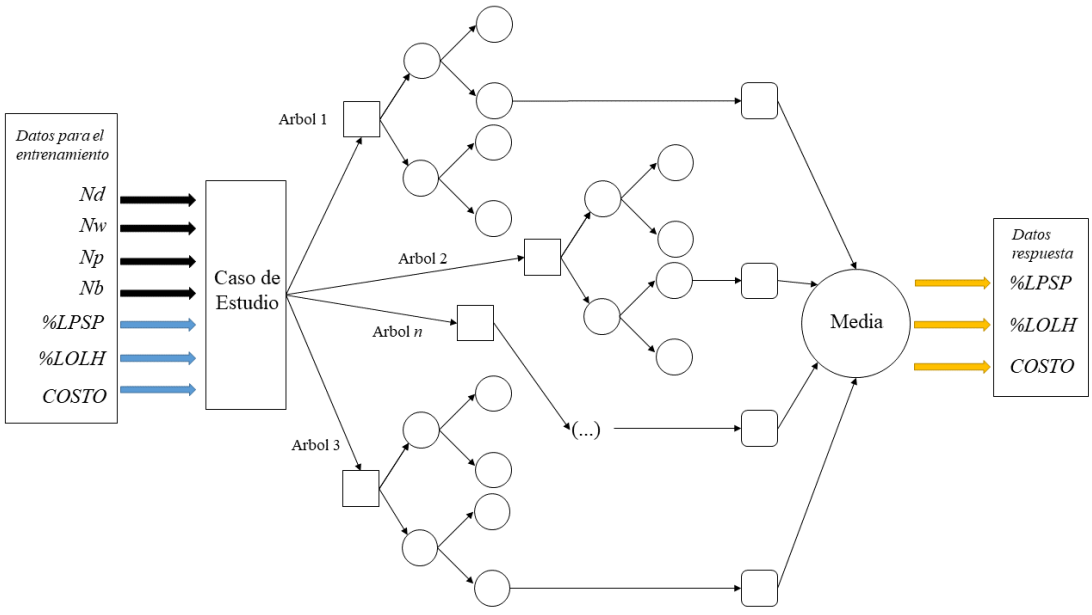
$$RSS = \sum_{j=1}^{J} \sum_{i \in Rj} \left(y_i - \hat{y}_{Rj}\right)^2 \tag{15}$$

Donde *J* son las regiones que minimizan el RSS, *Rj* son Regiones *(R1, R2, R3, ..., Rj)* y $\hat{y}_{Rj}$ es la media de la variable respuesta de la región *Rj*. Una vez comprendido el funcionamiento de los árboles de decisión, podemos describir el modelo de bosques aleatorios (Random forest).

2.2.3.2. Random Forest.

Un modelo de Random Forest está conformado por un conjunto de árboles de decisión, cada árbol selecciona muestras aleatorias realizando remuestreo en los datos de entrenamiento, esto significa que cada uno es entrenado con datos ligeramente distintos [12]. el árbol realiza observaciones en los datos de forma distribuida por cada nodo, hasta obtener una estructura final como se observa en la figura 6, para así mismo llegar a un nodo terminal que contiene el mejor resultado. Después de realizar este procedimiento, el modelo extrae el dicho valor de todos los árboles que lo conforman y obtiene la media de las predicciones (Figura 7).

Figura 5.
Ejemplo Random Forest en el problema de dimensionamiento de microrredes



Cada observación o nodo del modelo cuenta con variables de entrada o predictores *X* y variables de salida *Y*, de cada árbol se obtiene un nodo terminal el cual es seleccionado para extraer

el resultado final que contiene las variables *Y*, luego se suman y se divide por la cantidad *m* de variables resultantes y obtener la predicción del árbol *n.*

$$\hat{y}Arbol_n = \frac{Y_1 + Y_2 + \cdots + Y_m}{m} \tag{16}$$

La predicción final del modelo de Random Forest es la media de todas las predicciones individuales.

$$\hat{\mu} = \frac{\hat{y}Arbol_1 + \hat{y}Arbol_2 + \cdots + \hat{y}Arbol_n}{n} \tag{17}$$

2.2.3.3. Bagging.

El Bagging (en español, empaquetado) es un método de muestreo diseñado para obtener mayor estabilidad y precisión en los modelos de aprendizaje automático, esto es debido a que los modelos de M.L. tienden a sufrir problemas de equilibrio en el bias y varianza (Trevor Hastie, Robert Tibshirani, and Jerome Friedman 2009). El termino bias es utilizado para definir qué tan lejos están las predicciones de un modelo con respecto a sus valores reales, mientras que la varianza se refiere a cuanto cambia según los datos usados para el entrenamiento. Un modelo no debe cambiar demasiado su resultado debido a ligeras alteraciones en los datos utilizados, y en el caso de que así fuese, es porque está memorizando los datos en lugar de estudiarlos para hallar la relación entre los valores de entrada y su respectiva respuesta. Por esta razón, Random Forest utiliza el Bagging para obtener un bias y una varianza equilibrada y mejorar las predicciones debido a que evita la correlación de los árboles generados en el proceso, ajustando cada uno con un subconjunto distinto de datos de entrenamiento (Max Kuhn and Kjell Johnson 2013).

3. Propuesta

Las técnicas de regresión dentro del Machine Learning (M.L.) son de gran utilidad, ya que ayudan a optimizar problemas extensos. Se propone hacer uso de estas técnicas en el dimensionamiento de microrredes, con el fin de estudiar si son precisas y así ahorrar gran parte del tiempo que lleva la formulación tradicional.

3.1. Caso de estudio

De la tesis de maestría "Dimensionamiento de microrredes basada en Análisis de Ciclo de Vida" del ingeniero Iván Edgardo Jiménez Vargas (Iván Edgardo Jiménez Vargas 2021), se obtiene un set de datos a partir de una formulación clásica de dimensionamiento para condiciones climáticas y de carga específicas.

Tabla 1.
Elementos que componen la microrred para el caso de estudio

	Elementos que componen la microrred			
	Nd	Nw	Np	Nb
Cantidad mínima	1	0	0	1
Cantidad máxima	4	20	160	30
Configuraciones posibles	405720			

Donde *Nd* es el número de Generadores Diesel, *Nw* es el número de Turbinas eólicas, *Np* es el número de paneles solares y *Nb* es el número de baterías.

Para el cálculo de las configuraciones se tuvo en cuenta la cantidad mínima y máxima de cada unidad de generación y almacenamiento de energía.

$$Nd_{combi} = (Nd_{maxima} - Nd_{minima}) + 1 = (4 - 1) + 1 = 4 \quad (18)$$

$$Nw_{combi} = (Nw_{maxima} - Nw_{minima}) + 1 = (20 - 0) + 1 = 21 \quad (19)$$

$$Np_{combi} = (Np_{maxima} - Np_{minima}) + 1 = (160 - 0) + 1 = 161 \quad (20)$$

$$Nb_{combi} = (Nb_{maxima} - Nb_{minima}) + 1 = (30 - 1) + 1 = 30 \quad (21)$$

El + 1 indica que en cada grupo de generación y almacenamiento se incluye la cantidad mínima de elementos.

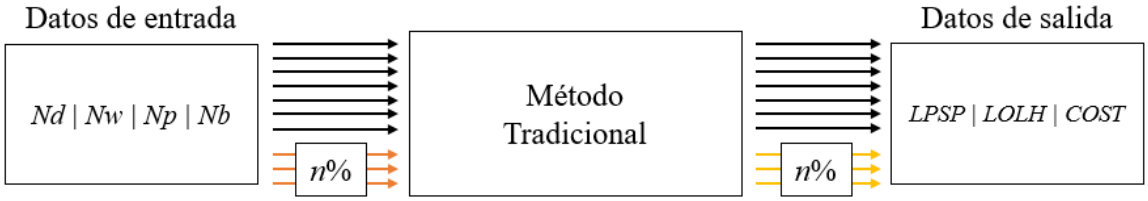
$$Configuraciones = Nd_{combi} * Nw_{combi} * Np_{combi} * Nb_{combi} \quad (22)$$

$$Configuraciones = 4 * 21 * 161 * 30 = 405720 \quad (23)$$

Considerando estas configuraciones como entradas, se identifica para cada una de estas tres salidas (Respuesta del problema de dimensionamiento): LPSP, LOLH y COSTO.

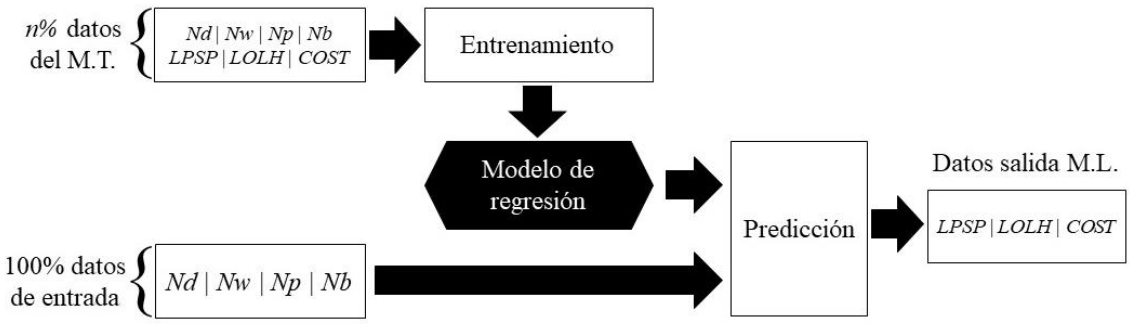
El método tradicional (M.T.) representado en la Figura 2, nos permite obtener los resultados de forma precisa, pero con un procesamiento de información bastante lento.

Figura 6.
Solución al problema de dimensionamiento de microrredes tradicionalmente



Las 3 técnicas de regresión de aprendizaje automático a experimentar serán la regresión lineal, la regresión polinómica y bosques aleatorios; con ellas se propone una solución al problema de dimensionamiento (figura 2) donde se escoge un *n*% de datos del M.T. (figura 1) para entrenar las técnicas de regresión. Con el modelo obtenido del entrenamiento, se aplica predicción al 100% de los datos de entrada, para obtener los datos de salida a partir de la predicción hecha por la técnica de regresión de M.L.

Figura 7.
Solución propuesta al problema de dimensionamiento de microrredes



4. Metodología

En este apartado se observa el *Data Set* con el fin de procesar y modelar los datos para extraer patrones e información que permita predecir valores que serán útiles en la etapa de análisis.

4.1. Código

Para el desarrollo de este proyecto se tienen en cuenta los datos obtenidos del código de Matlab descrito anteriormente (Iván Edgardo Jiménez Vargas 2021), donde se tiene un Data Set que cuenta con 405720 filas de datos. Cada fila cuenta con 7 columnas, las cuales son el número de generadores diésel, numero de generadores eólicos, numero de paneles solares, numero de baterías, la probabilidad de pérdida del suministro eléctrico (LPSP), las pérdidas de hora de carga (LOLH) y el costo (COST). Todas estas columnas están organizadas de izquierda a derecha respectivamente (tabla 1).

Tabla 2.
Dataset del caso de estudio

Nd	Nw	Np	Nb	LPSP	LOLH	COST
1	0	0	1	40,9854	89,3721	60.634,95
1	0	0	2	40,9854	89,3721	60.635,47
…	…	…	…	…	…	…
4	20	160	30	0,000024	0,002283	532871,57

Los métodos de entrenamiento mostrados en la metodología serán adaptados al Data Set, utilizando el lenguaje de programación Python, de aquí se define inicialmente tres códigos de entrenamiento, método de regresión lineal múltiple, método de regresión polinómica, y método de bosques aleatorios o “Random Forest”.

4.2. Selección de datos de entrenamiento

La selección de datos se realiza con el fin de determinar cómo se puede trabajar el Data Set y obtener los resultados de los métodos de regresión más cercanos al valor óptimo real.

Todos los métodos de regresión necesitan un porcentaje de datos para entrenamiento (training) y un porcentaje de datos para prueba (testing). La idea principal es que el porcentaje de los datos de entrenamiento sea el menor posible, siempre y cuando el resultado de los métodos de regresión permanezca cercano al resultado óptimo. La selección de datos se realizará con *Datos Aleatorios* y *Datos por salto*.

Para la selección de los datos, se utilizarán dos librerías: *Pandas* y *Scikit-learn*. *Pandas* permite leer el Data Set a partir de un documento con extensión *.csv* (el cual es el documento que almacena los datos) y así mismo guardar los resultados en *DataFrames* y exportarlos de nuevo en documentos con extensión *.csv* para utilízalos en el modelado (Nelli 2018).

Scikit-learn es una librería que se utiliza comúnmente en métodos de aprendizaje automático, para este caso se emplea en la selección de datos aleatorios (Scikit-learn developers 2022). De esta librería se llama al método *Model Selection* de donde se importa la función *train_test_split* que nos permite dividir nuestros datos en conjuntos de entrenamiento y prueba (Nelli 2018).

```
1. import pandas as pd
2. from sklearn.model_selection import train_test_split
```

4.2.1. Datos Aleatorios

La selección de los datos aleatorios se obtiene seleccionando los datos al azar de todo el Data Set, lo que significa que no sigue ningunas pautas definidas.

```
1. Xtrain, Xtest, Ytrain, Ytest = train_test_split(X,Y,test_size=n/1
   00)
```

Donde *X* y *Y* son los valores de entrada (*Nd*, *Nw*, *Np* y *Nb*) y salida (*LPSP*, *LOLH* y *COST*) del método, y *test_size* es la cantidad de datos a seleccionar para prueba (siendo 0 = 0% y 1 = 100%). La función guarda los resultados en 4 variables y luego son exportadas en Data Sets para entrenamiento y prueba.

4.2.2. Datos Por Salto

la selección de los datos con salto se obtiene estableciendo como punto inicial la primera posición de todo el Data Set, y a partir de ahí seleccionar un dato cada enésima cantidad de datos.

Para este código se utiliza un bucle *For* que evalúa los datos cada enésimo paso y junto al método ILOC de la librería Pandas que permite realizar selecciones por posición en un DataFrame, se van guardando en una variable que luego es exportada como Data Set de entrenamiento, mientras que los valores restantes que no fueron seleccionados son guardados en una variable que es exportada como Data Set de prueba (NumFOCUS 2022).

Al momento de realizar el entrenamiento de una técnica de regresión, es importante tener un buen número de pruebas realizadas, para así en un futuro tener sustento y poder elegir un porcentaje de entrenamiento lo más bajo sea posible. En la tabla 3 podemos observar los porcentajes que se van a trabajar para cada uno de los métodos de selección de datos.

Tabla 3.
Métodos para la selección de datos

Método de selección de datos	Datos Aleatorios							Datos por Salto						
Porcentajes (%)	1	5	10	20	33	50	100	1	5	10	20	33	50	100

En la tabla 4, se observa el tamaño de las entradas y salidas para cada porcentaje de entrenamiento.

Tabla 4.
Cantidad de datos de entrada y salida

% de entrenamiento	X_train (Nd, Nw, Np, Nb)	Y_train (LPSP, LOLH, COST)
100	405720 x 4	405720 x 3
50	202860 x 4	202860 x 3
33	133887 x 4	133887 x 3
20	81144 x 4	81144 x 3
10	40572 x 4	40572 x 3
5	20286 x 4	20286 x 3
1	4057 x 4	4057 x 3

Podemos determinar el tamaño del salto en el Data Set utilizando el porcentaje de datos en la siguiente formula:

$$\frac{100}{n} = S \tag{24}$$

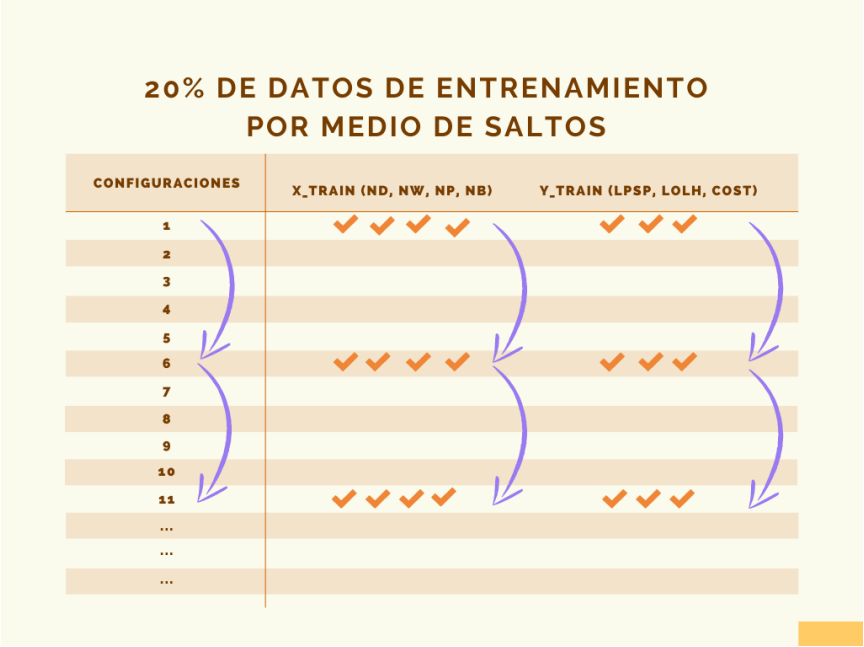
Donde n es el porcentaje de datos y S el espacio de datos por salto. Con la información anterior, determinamos los saltos de cada uno de los porcentajes de la tabla 11.

Tabla 5.
Selección de datos de entrenamiento por medio de saltos

Cantidad de datos por salto							
n	1	5	10	20	33	50	100
S	100	20	10	5	3	2	1

Por ejemplo, para la selección de datos por medio de saltos para el 20% de entrenamiento, se toma una configuración cada 5 posiciones

Figura 8.
Metodología para la selección de datos de entrenamiento por medio de saltos



Cada uno de los métodos y los porcentajes se exportan en cuatro archivos con extensión .csv los cuales son:

Xtrain.csv: datos de entrada para entrenamiento.

Xtest.csv: datos de entrada para prueba.

Ytrain.csv: datos de salida para entrenamiento.

Ytest.csv: datos de salida para prueba.

Estos cuatro archivos se obtendrán de cada porcentaje de datos de entrenamiento seleccionado, los cuales se extraen de cada método de selección de datos; esto da un total de cincuenta y seis (56) archivos donde cada grupo de cuatro archivos será almacenado en carpetas distintas para reconocer a que método de selección de datos y porcentaje pertenece; con el fin de evaluar en cada uno de los métodos de regresión de aprendizaje automático. En la figura 10, se

pueden observar los cuatro archivos mencionados anteriormente, donde su ruta está en *"…/Datos_Aleatorios/Datos_20"*; es decir, que estos archivos representan un porcentaje de datos de entrenamiento del 20%, utilizando el método de selección de datos aleatorios.

Figura 9.
Ejemplo de ubicación de los archivos con los datos de entrenamiento y testeo

4.3. Construcción del código

Todos los códigos requieren de las mismas librerías presentadas a continuación:

```
1. from google.colab import drive
2. import pandas as pd
3. import numpy as np
4. from sklearn.model_selection import train_test_split
5. import matplotlib.pyplot as plt
6. import seaborn as sns
```

Google.colab es un servicio cloud que nos ofrece google para poder trabajar códigos basados en Notebooks de Jupyter, estas Notebooks trabajan en Python 2.7 y 3.6, para este caso, ese servicio es llamado con el fin de importar *Drive*, el cual nos permite habilitar el acceso a las carpetas y archivos de Google Drive utilizando el siguiente código:

```
1. drive.mount('/content/drive')
```

Esto es realizado con el fin de trabajar los códigos en la nube, incluyendo desde la selección de datos, la construcción de los códigos de regresión, y la exportación de diagramas, en carpetas compartidas de Google Drive.

Las librerías *Pandas* y *SkLearn* fueron mencionadas en la sección 3.2. Por ello se mencionará aquellas librerías nuevas con las que se desarrollan los códigos de regresión.

Numpy es una librería que se especializa en el cálculo numérico y análisis de datos, principalmente para una gran cantidad de datos (Nelli 2018). *Numpy* presenta una nueva métodología para trabajar los *Arrays*, ya que el procesamiento de *Arrays* con *Numpy* es 50 veces más rápido que los *Arrays* predefinidos en Python lo que hace que esta librería sea ideal para procesar matrices de grandes dimensiones.

Matplotlib es una librería que se especializa en crear gráficos de dos dimensiones, se suele utilizar para gráficar grandes cantidades de datos, además de permitir crear y personalizar distintos tipos de gráficos comunes (John Hunter et al. 2022).

Seaborn es una librería basada en *Matplotlib* que permite dibujar gráficos estadísticos informativos.

4.3.1. Configuración Óptima

La configuración óptima es determinada a partir del Data Set original, para ello se debe filtrar aquellos valores de LPSP y LOLH que sean menores a 2.5% y 5% respectivamente.

```
1. filterRealDataset = dataset[(dataset.LPSP < maxLPSP) & (dataset.L
   OLH < maxLOLH)]
```

Luego de filtrar esos datos, por medio de la librería Pandas, se usa el método *idxmin()* (NumFOCUS 2022), el cual permite hallar la **posición** del valor mínimo de una columna en un Data Set. En este caso, se realiza la búsqueda en la columna COST del Data Set filtrado *filterRealDataset* y también su valor numérico. En dicha columna.

```
1. print("Posicion:    ", filterRealDataset.COST.idxmin()+1,"\nCOSTO:
   ", filterRealDataset.COST.min())
```

La función *print()* se utiliza para mostrar en pantalla diferentes variables separadas por comas para combinar texto con variables numéricas.

De esta última línea de código se obtiene el siguiente resultado (tabla 12), el cual es la configuración óptima utilizada para comparar con la mejor configuración de cada uno de los métodos de regresión evaluados.

Tabla 6.
Configuración óptima para el Data Set del caso de estudio

Configuración óptima	108552
Nd	2
Nw	1
Np	76
Nb	12
LPSP %	1,24
LOLH %	4,95
COST (US$)	115887,03

4.3.2. Desarrollo de programación de los métodos de regresión

Para la construcción de cualquiera de los 3 métodos de regresión desarrollados, se utiliza la librería *Pandas* para la selección de los siguientes archivos con extensión *.csv*:

- *Datos.csv:* este archivo contiene las 405720 filas y 7 columnas de datos.
- *Xtrain.csv*: datos de entrada para entrenamiento.
- *Xtest.csv*: datos de entrada para prueba.
- *Ytrain.csv*: datos de salida para entrenamiento.
- *Ytest.csv*: datos de salida para prueba.

Los últimos 4 archivos son dependientes del porcentaje de entrenamiento y de prueba seleccionado para la ejecución del código. Por ejemplo, si se desea trabajar con 5% de datos aleatorios, se importan los elementos de la carpeta que contiene los 4 archivos realizando el

procedimiento de la sección 3.2. Luego, cada uno de estos archivos es guardado en una variable, con el fin de facilitar distintos procedimientos del entrenamiento.

4.3.2.1. Regresión Lineal.

Para realizar el entrenamiento utilizando el método de regresión Lineal, es necesaria la siguiente función:

```
1. from sklearn import linear_model
```

Linear_model es una clase de la librería Sklearn el cual contiene distintas funcionalidades para el aprendizaje automático con modelos lineales (Scikit-learn developers 2022). Por medio de los datos suministrados, obtiene automáticamente una "recta" con respecto a la predicción buscada.

```
1. lr_multiple = linear_model.LinearRegression()
```

El método de regresión es guardado en la variable *lr_multiple* para poder utilizar el objeto *fit( )* y el objeto *predict( )*.

El objeto *fit( )* nos permite implementar los datos de entrenamiento al método de regresión.

Para hallar la variable de salida Y se aplica *predict( )*, que recibe todos los datos de prueba para realizar la predicción a partir del modelo entrenado con los parámetros de aprendizaje de *fit( )*, donde obtenemos la variable *y_pred* que contiene los datos de salida obtenidos de la predicción.

```
1. Y_pred = lr_multiple.predict(X_test)
```

La variable *y_pred* es utilizada para determinar el error relativo con respecto a los datos reales, y dibujar distintos mapas de calor para determinar si los valores de LPSP y LOLH se encuentran en valores porcentuales menores a 2.5% y 5% respectivamente.

Como sabemos, una función lineal es aquella cuya expresión algebraica es:

$$Y = ax + b \tag{25}$$

Donde *a* es el valor de la pendiente de la recta y *b* es la intersección con el eje vertical *Y*. La libreria Sklearn cuenta con estos dos atributos importantes en el caso que sea requerido conocer el modelo matemático obtenido del entrenamiento:

- *coef_*: Permite obtener los valores de las pendientes o coeficientes *a*.
- *intercept_*: Obtiene los valores de la intersección o coeficientes *b*.

```
1. print(lr_multiple.coef_)
2. print(lr_multiple.intercept_)
```

4.3.2.2. Regresión Polinómica.

Realizaremos el entrenamiento con el método de regresión Polinómica utilizando las siguientes funciones:

```
1. from sklearn.pipeline import make_pipeline
2. from sklearn.preprocessing import PolynomialFeatures
3. from sklearn.preprocessing import StandardScaler
4. from sklearn.linear_model import LinearRegression
```

Pipeline es una clase de la librería *Sklearn* (Scikit-learn developers 2022). Esta clase está diseñada para realizar transformaciones de datos con el fin de crear un flujo de trabajo coherente. Esta clase es bastante utilizada en métodos de M.L. para encadenar varias transformaciones juntas y terminar con una estimación de algún tipo.

PolynomialFeatures y *StandardScaler* son clases de preprocesamiento de la librería *Sklearn*. *PolynomialFeatures* está diseñada para recibir como entrada distintas características predictivas y las transforma en posibles configuraciones hasta el grado polinómico indicado (Sebastian Raschka and Vahid Mirjalili 2017). *StandarScaler* se encarga de seguir la distribución normal estándar (SND) es decir, escala los datos en un rango de 0 a 1 para valores positivos o entre -1 y 1 cuando existen valores negativos; esto se hace para una mayor optimización del pipeline donde se trabajan estas dos últimas clases en conjunto al método de regresión polinómica.

El método *LinearRegression* es utilizado en el pipeline debido a que la regresión polinómica parte del modelo lineal, donde se agrega curvatura elevando las variables independientes (sección 2.2.2).

A continuación, la línea de código donde definimos el grado del polinomio en la clase *PolynomialFeatures*:

```
1. poli_reg =make_pipeline(PolynomialFeatures(3),StandardScaler(),Li
   nearRegression())
```

El *Pipeline* es guardado en la variable *poli_reg*, y a continuación aplicamos los objetos *fit( )* y *Predict( )* mencionados en la sección 3.3.2.1. para realizar el entrenamiento del modelo.

```
1. poli_reg.fit(X_train_p, y_train_p)
2. y_pred = poli_reg.predict(X_test_p)
```

La variable *y_pred* es utilizada para determinar el error relativo con respecto a los datos reales, y dibujar distintos mapas de calor para determinar si los valores de LPSP y LOLH se encuentran en valores porcentuales menores a 2.5% y 5% respectivamente.

4.3.2.3. Random Forest.

Para empezar con el entrenamiento utilizando el método de regresión Random Forest, es necesaria la siguiente función:

```
1. from sklearn.ensemble import RandomForestRegressor
```

RandomForestRegressor es un estimador que permite ajustar distintos arboles de decisión para un conjunto de datos, y utiliza un promedio para mejorar la precisión predictiva (sección 2.2.3) (Sebastian Raschka and Vahid Mirjalili 2017). Esta función nos permite modificar distintos parámetros como por ejemplo el número de árboles, criterios de aleatoriedad, y distintas características para buscar la mejor condición.

```
1. regressor = RandomForestRegressor(n_estimators=100, random_state=
   123)
```

El método de regresión es guardado en la variable *regressor*, donde la cantidad de árboles seleccionados para el entrenamiento son 100 y el valor de aleatoriedad es 123.

Se seleccionaron 100 árboles debido a que una mayor cantidad representa un mayor consumo de TPU (Unidad de Procesamiento Tensorial) lo que requiere mayor tiempo para

procesar la información, mientras que una menor cantidad de árboles arrojan resultados deficientes para su análisis a profundidad al realizar la comparativa con los otros métodos.

El valor de aleatoriedad es una semilla que permite controlar que tan aleatorio son los resultados cada vez que se ejecuta esta línea de código; para este caso, con la semilla 123 estamos indicando que en cualquier ejecución se va a obtener el mismo resultado. Esto se hace para que los resultados obtenidos en la primera ejecución no se pierdan al volver a ejecutar el código debido a distintas modificaciones, como agregar un gráfico o imprimir en pantalla una variable en específico para observar su comportamiento.

Se deben aplicar los archivos *.csv* a la variable de *regressor* para realizar el entrenamiento de los datos en Random Forest utilizando el objeto *fit( )* y el objeto *predict( )* mencionados en la sección 3.3.2.1.

```
1. regressor.fit(X_train, y_train)
2. y_pred = regressor.predict(X_test)
```

La variable *y_pred* es utilizada para determinar el error relativo con respecto a los datos reales, y dibujar distintos mapas de calor para determinar si los valores de LPSP y LOLH se encuentran en valores porcentuales menores a 2.5% y 5% respectivamente.

4.3.3. Precisión de un modelo de regresión

Es importante conocer que tan bien funcionan los diferentes algoritmos en un conjunto de datos en particular. La librería *Sklearn* cuenta con un comando que ayudar a evaluar la precisión de un modelo de regresión comparando los datos de prueba con la técnica una vez realizado el entrenamiento (Scikit-learn developers 2022).

```
1. Precision = algoritmo.score(X_test, y_test))
```

El coeficiente R^2 el cual está definido en el comando *score* que indica qué tan preciso es el modelo siendo cercano a 1 máxima precisión y a 0 una precisión no deseada.

$$R^2 = 1 - \frac{ESS}{TSS} \tag{26}$$

4.3.3.1. ESS (Suma residual de cuadrados).

En estadística e inteligencia artificial, la suma residual de cuadrados (RSS por sus siglas en inglés) es una medida de la discrepancia entre los datos de un modelo de (Kevin P. Murphy 2012).

$$RSS = \sum_{i=1}^{n} (y_i - f(x_i))^2 \tag{27}$$

Donde y_i es la salida real al problema y $f(x_i)$ es la predicción obtenida del modelo.

4.3.3.2. TSS (Suma total de cuadrados).

En estadística, la suma total de cuadrados (TSS o SST) se define como la suma de todas las diferencias al cuadrado entre sus datos reales y su media global (Archdeacon 1994).

$$TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2 \tag{28}$$

Donde y_i es la salida real al problema y $\bar{y}$ es el promedio de este conjunto de datos

Por consiguiente:

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - f(x_i))^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2} \tag{29}$$

4.4. Resultados del método propuesto

A continuación, se presentan los resultados obtenidos de cada técnica con sus respectivos grupos de datos de entrenamiento. En las siguientes tablas *Config. óptima* hace referencia a la solución real del problema indicada anteriormente (tabla 6). *Posición* indica el número de la configuración que cumple con los requisitos para ser la óptima, ya sea de la solución real o en cada una de las técnicas. *Config. Reg. Lin.*, *Config. Reg. Poli.*, y *Config. Random Forest* hacen referencia a los resultados obtenidos al entrenar y evaluar las técnicas y *Config. Dataset*, representa las salidas reales obtenidas del Dataset original en la posición indicada.

Tabla 7.

*Resultados regresión lineal, datos de entrenamiento escogidos de manera **aleatoria***

	Config. óptima	Config. Reg. Lin.	Config. Dataset	Config. Reg. Lin.	Config. Dataset	Config. Reg. Lin.	Config. Dataset	Config. Reg. Lin.	Config. Dataset	Config. Reg. Lin.	Config. Dataset	Config. Reg. Lin.	Config. Dataset	Config. Reg. Lin.	Config. Dataset
Porcentaje entren.		1%		5%		10%		20%		33%		50%		100%	
Posición	108552	306181		306271		306151		306121		306211		306211		306181	
Nd	2	4		4		4		4		4		4		4	
Nw	1	0		0		0		0		0		0		0	
Np	76	63		66		62		61		64		64		63	
Nb	12	1		1		1		1		1		1		1	
LPSP %	1,24	1,23	0,12	1,27	0,12	1,25	0,12	1,24	0,12	1,26	0,12	1,25	0,12	1,26	0,12
LOLH %	4,95	4,98	2,45	4,98	2,43	4,95	2,45	4,98	2,46	4,97	2,44	4,97	2,44	4,99	2,45
COST (US$)	115887,03	82311,08	128174,12	83124,15	128587,41	82196,33	128087,16	82341,34	127998,94	83110,32	128299,49	83066,14	128299,49	82799,5	128174,12

Tabla 8.

*Resultados Regresión lineal, datos de entrenamiento escogidos por medio de **saltos***

	Config. óptima	Config. Reg. Lin.	Config. Dataset	Config. Reg. Lin.	Config. Dataset	Config. Reg. Lin.	Config. Dataset	Config. Reg. Lin.	Config. Dataset	Config. Reg. Lin.	Config. Dataset	Config. Reg. Lin.	Config. Dataset	Config. Reg. Lin.	Config. Dataset
Porcentaje entren.		1%		5%		10%		20%		33%		50%		100%	
Posición	108552	306421		306391		306391		306301		306341		306211		306181	
Nd	2	4		4		4		4		4		4		4	
Nw	1	0		0		0		0		0		0		0	
Np	76	71		70		70		67		65		64		63	
Nb	12	1		1		1		1		1		1		1	
LPSP %	1,24	1,2	0,12	1,19	0,12	1,19	0,12	1,22	0,12	1,24	0,12	1,25	0,12	1,26	0,12
LOLH %	4,95	4,97	2,42	4,97	2,42	4,97	2,42	4,96	2,43	4,98	2,44	4,99	2,44	4,99	2,45
COST (US$)	115887,03	87338,17	129312,77	87082,72	129163,18	87086,36	129163,18	84931,12	128730,85	83870,9	128432,51	83332,81	128299,49	82810,99	128174,12

Tabla 9.

*Resultados regresión polinómica, datos de entrenamiento escogidos de manera **aleatoria***

	Config. óptima	Config. Reg. Poli.	Config. Dataset	Config. Reg. Poli.	Config. Dataset	Config. Reg. Poli.	Config. Dataset	Config. Reg. Poli.	Config. Dataset	Config. Reg. Poli.	Config. Dataset	Config. Reg. Poli.	Config. Dataset	Config. Reg. Poli.	Config. Dataset
Porcentaje entren.		1%		5%		10%		20%		33%		50%		100%	
Posición	108552	205721		205751		205721		205661		205691		205900		205691	
Nd	2	3		3		3		3		3		3		3	
Nw	1	0		0		0		0		0		0		0	
Np	76	95		96		95		93		94		101		94	
Nb	12	11		11		11		11		11		10		11	
LPSP %	1,24	0,98	0,25	1	0,24	1,05	0,25	1,04	0,26	1,03	0,25	1,03	0,3	1,03	0,25
LOLH %	4,95	4,97	2,52	4,98	2,49	4,98	2,52	4,99	2,58	4,97	2,55	4,97	3,15	4,99	2,55
COST (US$)	115887,03	121228,64	128360,57	121772,52	128604,24	121300,83	128360,57	121201,24	127889,8	121336,61	128116,76	121154,63	128806,71	121266,43	128116,76

Tabla 10.

*Resultados Regresión polinómica, datos de entrenamiento escogidos por medio de **saltos***

	Config. óptima	Config. Reg. Poli.	Config. Dataset	Config. Reg. Poli.	Config. Dataset	Config. Reg. Poli.	Config. Dataset	Config. Reg. Poli.	Config. Dataset	Config. Reg. Poli.	Config. Dataset	Config. Reg. Poli.	Config. Dataset	Config. Reg. Poli.	Config. Dataset
Porcentaje entren.		1%		5%		10%		20%		33%		50%		100%	
Posición	108552	205572		205602		205602		205781		205751		205721		205691	
Nd	2	3		3		3		3		3		3		3	
Nw	1	0		0		0		0		0		0		0	
Np	76	90		91		91		97		96		95		94	
Nb	12	12		12		12		11		11		11		11	
LPSP %	1,24	0,98	0,23	0,98	0,22	0,98	0,22	1,02	0,24	1,03	0,24	1,03	0,25	1,04	0,25
LOLH %	4,95	4,99	2,19	4,95	2,14	4,95	2,14	4,99	2,47	4,97	2,49	4,98	2,52	4,99	2,55
COST (US$)	115887,03	122671,04	128121,1	122752,84	128341,11	122747,7	128341,11	121947,76	128851,25	121672,39	128604,24	121452,24	128360,57	121266,45	128116,76

Tabla 11.

*Resultados Random forest, datos de entrenamiento escogidos de manera **aleatoria***

	Config. óptima	Config. Random Forest	Config. Dataset	Config. Random Forest	Config. Dataset	Config. Random Forest	Config. Dataset	Config. Random Forest	Config. Dataset	Config. Random Forest	Config. Dataset	Config. Random Forest	Config. Dataset	Config. Random Forest	Config. Dataset
Porcentaje entren.		1%		5%		10%		20%		33%		50%		100%	
Posición	108552	213728		108524		108672		108582		108493		108672		108552	
Nd	2	3		2		2		2		2		2		2	
Nw	1	2		1		1		1		1		1		1	
Np	76	40		75		80		77		74		80		76	
Nb	12	2		14		12		12		13		12		12	
LPSP %	1,24	0,64	0,62	1,32	1,18	1,2	1,12	1,24	1,21	1,28	1,25	1,19	1,12	1,25	1,24
LOLH %	4,95	4,88	4,78	4,96	4,44	4,91	4,43	4,96	4,85	4,92	4,81	4,9	4,53	4,97	4,95
COST (US$)	115887,03	120384,87	117486,84	117131,77	117183,28	116822,84	116901,14	115996,31	116141,77	116238,4	116377,26	116297,98	116901,14	115845	115887,03

Tabla 12.

*Resultados Random forest, datos de entrenamiento escogidos por medio de **saltos***

	Config. óptima	Config. Random Forest	Config. Dataset	Config. Random Forest	Config. Dataset	Config. Random Forest	Config. Dataset	Config. Random Forest	Config. Dataset	Config. Random Forest	Config. Dataset	Config. Random Forest	Config. Dataset	Config. Random Forest	Config. Dataset
Porcentaje entren.		1%		5%		10%		20%		33%		50%		100%	
Posición	108552	213811		108757		108727		108729		108462		105463		105552	
Nd	2	3		2		2		2		2		2		2	
Nw	1	2		1		1		1		1		1		1	
Np	76	43		83		82		82		73		73		76	
Nb	12	1		7		7		9		12		13		12	
LPSP %	1,24	0,64	0,63	1,17	2,07	1,18	2,08	1,18	1,52	1,3	1,35	1,3	1,29	1,25	1,24
LOLH %	4,95	4,99	4,94	4,98	8,63	4,99	8,65	4,99	6,63	4,98	5,3	4,98	4,95	4,96	4,95
COST (US$)	115887,03	117944,35	117406,87	116558,03	112819,22	116476,85	112536,3	116493,47	114524,01	116105,86	115172,17	116103,75	116145,29	115862,03	115887,03

4.5. **Precisión de los modelos empleados**

Por medio del coeficiente R^2 score, se calcula la precisión de cada técnica de regresión con sus respectivos porcentajes de entrenamiento y método de selección de datos (aleatorio y saltos). En las siguientes tablas se puede apreciar la precisión de cada técnica.

Tabla 13.
Precisión de los modelos empleados con sus respectivos datos de entrenamiento escogidos de forma aleatoria

Selección de datos aleatorio			
Porcentaje de entrenamiento	Precisión		
	Regresión Lineal	Regresión Polinómica	Random Forest
1%	0.6312	0.9071	0.9495
5%	0.6316	0.9075	0.9882
10%	0.6311	0.9077	0.9948
20%	0.6311	0.908	0.9975
33%	0.6314	0.9076	0.9987
50%	0.6344	0.909	0.9991
100%	0.6361	0.9124	0.9993

Tabla 14.
Precisión de los modelos empleados con sus respectivos datos de entrenamiento escogidos por medio de saltos

Selección de datos por medio de saltos			
Porcentaje de entrenamiento	Precisión		
	Regresión Lineal	Regresión Polinómica	Random Forest
1%	0.6228	0.8923	0.984
5%	0.6216	0.8899	0.9887
10%	0.6193	0.8883	0.9885
20%	0.6301	0.9075	0.9978
33%	0.6311	0.9078	0.9992
50%	0.6313	0.9079	0.9993
100%	0.6314	0.9079	0.9999

5. Análisis

Una vez obtenida la mayor información posible en las etapas anteriores, se presentan tablas del error relativo porcentual, mapas de calor para la visualización de datos y gráficas de precisión (R^2 score) de los modelos implementados, con el fin de evaluar el impacto que tienen las técnicas de regresión en el dimensionamiento de microrredes.

5.1. Resultados obtenidos de los métodos de regresión

5.1.1. Regresión Lineal

Al entrenar con la totalidad de los datos, la regresión lineal no se acerca a la solución, esto se debe a que el dimensionamiento de microrredes no es un problema lineal. En las tablas 13 y 14 se aprecia que esta técnica prioriza el número de paneles solares, pero ignora por completo la cantidad de baterías. En una microrred, es indispensable el uso de baterías ya que, al almacenar energía en estas, se crea un mayor grado de autonomía energética. Adicionalmente, las baterías ayudan a equilibrar el sistema cuando la energía demandada supera la capacidad de generación.

En el caso de estudio se cuenta con un máximo de 4 generadores diesel y la regresión lineal recomienda hacer uso de todos estos. La configuración dada por la regresión lineal se aleja aproximadamente 200.000 posiciones de la configuración óptima. Al realizar la búsqueda de esta configuración en la regresión lineal (LPSP menor al 2.5% y LOLH menor del 5%), se observa que el costo es totalmente diferente a la solución real del problema y por consiguiente se hace la comparación de estos indicadores en la configuración deducida por esta técnica (Config. Reg. Lin y Config. Dataset).

5.1.2. Regresión Polinómica

La respuesta de la regresión polinómica al problema de dimensionamiento de microrredes mejora con respecto a la regresión lineal. Aunque el número baterías muestra un mejor comportamiento y la cantidad de paneles aumenta, esta técnica se encuentra lejos de la configuración óptima debido a que no responde como debe ser en cuanto al número generadores diesel y aerogeneradores.

La configuración resultante de la regresión polinómica se aleja aproximadamente 100.000 posiciones con respecto a la configuración óptima, esto se debe a que esta técnica recomienda hacer uso de 3 generadores diesel a diferencia de la solución real que demanda 2 de estas unidades. La regresión polinómica presenta un mejor costo y en las tablas 15 y 16 se puede observar el LPSP, LOLH y COST en la configuración recomendada por esta técnica, por consiguiente, se hace la comparación de estos indicadores (Config. Reg. Poli y Config. Dataset).

5.1.3. Random Forest

En el Random Forest, a diferencia de la regresión lineal y polinómica, el método de selección de datos de entrenamiento de forma aleatoria presenta una menor precisión que por medio de saltos, sin embargo, en las tablas 17 y 18 se observa que esta es la técnica que mas se acopla al problema de dimensionamiento de microrredes.

Al escoger mas del 5% de datos para el entrenamiento de forma aleatoria, se acerca a la solución del problema manteniendo la naturaleza de la composición de la microrred en su totalidad. Por el contrario, al escoger los datos de entrenamiento menor o igual al 20% por medio de saltos, se aleja de su composición real teniendo grandes diferencias en cuanto al número de paneles solares y baterias. En las tablas 17 y 18 se realiza la comparacion de los indicadores de confiabilidad y costo en las configuraciones recomendadas por el Random Forest (Config. Random Forest y Config. Dataset).

5.2. **Error relativo porcentual**

El error relativo porcentual es una medida que nos indica el grado de incertidumbre de una magnitud calculada con respecto a su valor real. Se calcula con la siguiente expresión:

$$Er\% = \frac{Valor\ real - Valor\ calculado}{Valor\ real} * 100 \tag{30}$$

A continuación, se presentan las tablas del error relativo porcentual para el LPSP, LOLH y COST donde el valor real está dado por *Config. óptima* y el valor calculado por *Config. Reg. Lin*, *Com. Reg. Poli* y *Com. Random Forest* de las tablas 7-12.

Tabla 15.
Error relativo porcentual para todas las técnicas de regresión, datos escogidos de forma ***aleatoria***

	ERROR RP LPSP			ERROR RP LOLH			ERROR RP COST		
% Entrena.	Reg. Lin	Reg. Poli	Random Forest	Reg. Lin	Reg. Poli	Random Forest	Reg. Lin	Reg. Poli	Random Forest
1	0,81	20,97	48,39	0,61	0,40	1,41	28,97	4,61	3,88
5	2,42	19,35	6,45	0,61	0,61	0,20	28,27	5,08	1,07
10	0,81	15,32	3,23	0,00	0,61	0,81	29,07	4,67	0,81
20	0,00	16,13	0,00	0,61	0,81	0,20	28,95	4,59	0,09
33	1,61	16,94	3,23	0,40	0,40	0,61	28,28	4,70	0,30
50	0,81	16,94	4,03	0,40	0,40	1,01	28,32	4,55	0,35
100	1,61	16,94	0,81	0,81	0,81	0,40	28,55	4,64	0,04

Tabla 16.
Error relativo porcentual para todas las técnicas de regresión, datos escogidos por medio de ***saltos.***

	ERROR RP LPSP			ERROR RP LOLH			ERROR RP COST		
% Entrena.	Reg. Lin	Reg. Poli	Random Forest	Reg. Lin	Reg. Poli	Random Forest	Reg. Lin	Reg. Poli	Random Forest
1	3,23	20,97	48,39	0,40	0,81	0,81	24,64	5,85	1,78
5	4,03	20,97	5,65	0,40	0,00	0,61	24,86	5,92	0,58
10	4,03	20,97	4,84	0,40	0,00	0,81	24,85	5,92	0,51
20	1,61	17,74	4,84	0,20	0,81	0,81	26,71	5,23	0,52
33	0,00	16,94	4,84	0,61	0,40	0,61	27,63	4,99	0,19
50	0,81	16,94	4,84	0,81	0,61	0,61	28,09	4,80	0,19
100	1,61	16,13	0,81	0,81	0,81	0,20	28,54	4,64	0,02

5.3. Mapas de calor para cada porcentaje de datos de entrenamiento

A continuación, se observa el comportamiento de cada método de aprendizaje automático utilizando mapas de calor, estos muestran el valor de LPSP y LOLH que se encuentran en un rango menor al 2.5% y 5% respectivamente, tanto en el Data Set original (Primera columna a la izquierda) como en la predicción obtenida en cada uno de los porcentajes de entrenamiento con su respectivo método de selección de datos.

5.3.1. LPSP

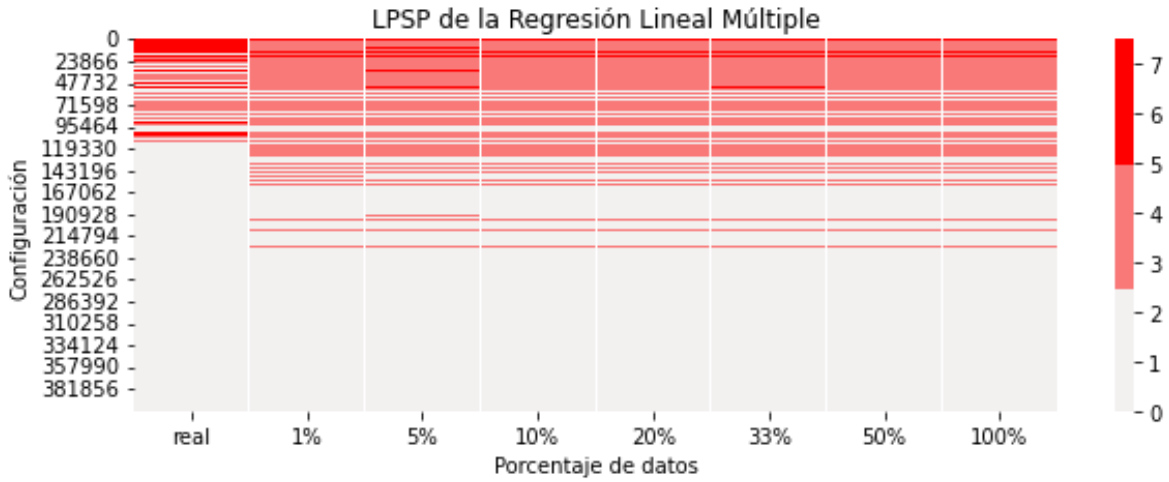
Los mapas de calor de esta sección muestran los resultados del LPSP para los métodos de selección de datos aleatorios y por saltos. En la barra de la derecha, se observan los colores de la gráfica, donde el color blanco indica los valores en un rango menor o igual a 2.5%, el color rosado indica los valores que se encuentran entre un 2.5% y 5%, y el color rojo que indica los valores superiores a 5%.

5.3.1.1. Datos Aleatorios.

En el mapa de calor de la figura 10, observamos un comportamiento en el que los datos no logran ajustarse al modelado lineal. A pesar de aumentar su porcentaje de entrenamiento de 1% a 100%, no se obtiene un resultado cercano al indiciador LPSP real, lo que significa que este método de regresión no puede presentar una solución precisa.

Figura 10.

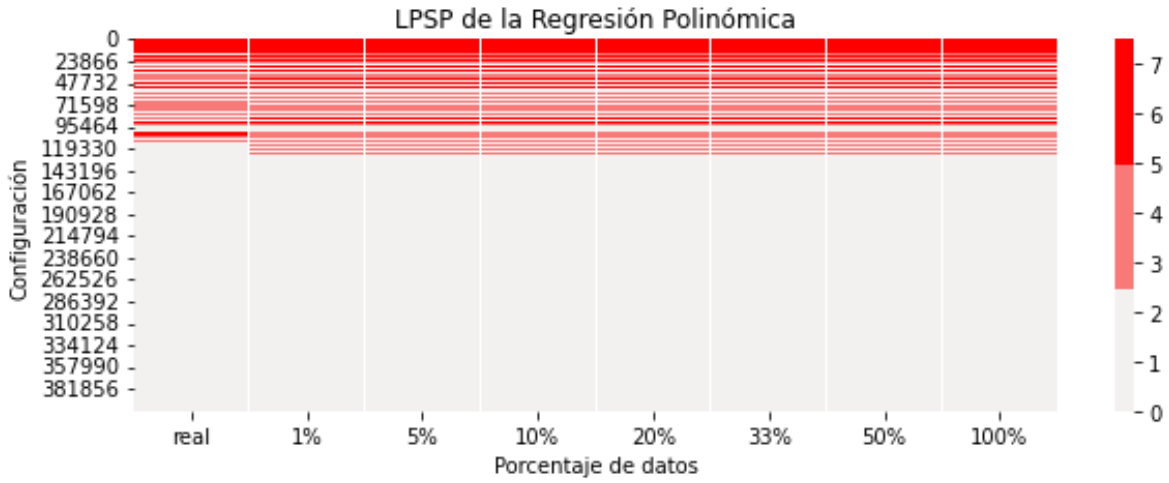
*Mapa de calor del LPSP para la regresión lineal múltiple (método de selección de datos de entrenamiento **aleatorio**)*



La regresión polinómica evidencia tener un resultado más acertado en el mapa de calor de la figura 11. Aunque su precisión no es superior al 92%, cambia levemente en cada uno de los porcentajes de entrenamiento empleados (tabla 13) lo que nos da a entender que este método logra ajustar bien un modelo matemático que represente el sistema a pesar de tener un porcentaje de datos bajo.

Figura 11.

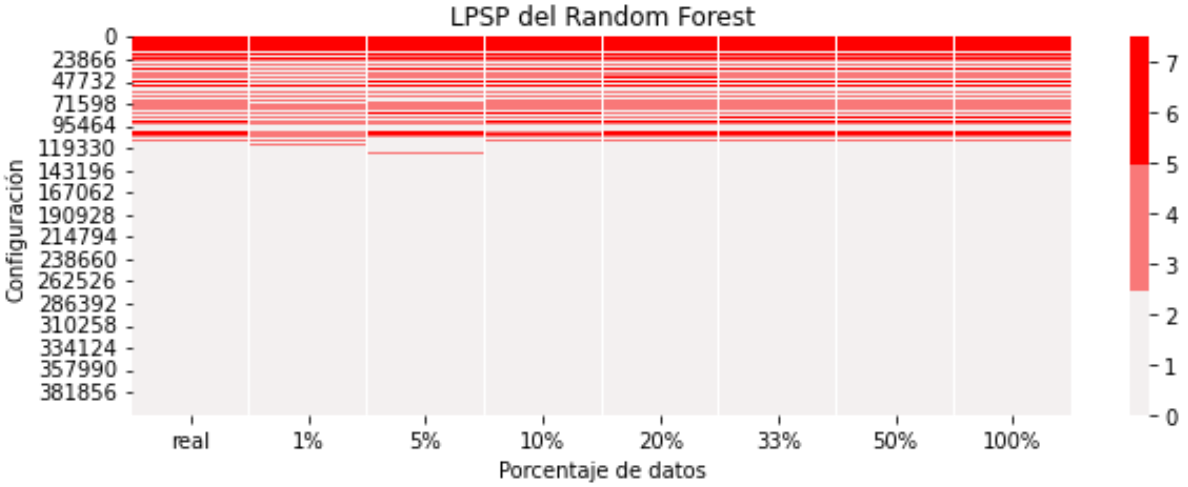
*Mapa de calor del LPSP para la regresión polinómica (método de selección de datos de entrenamiento **aleatorio**)*



La regresión de Random Forest logra tener unos resultados muy cercanos a los del Data Set original, como se observa en la figura 12, donde se reduce hasta el 5% los datos de entrenamiento, manteniendo una precisión superior al 98%. Entrenar con un porcentaje menor reduce considerablemente la precisión, pero para este caso, en el 1% de los datos, su precisión es mayor que la precisión máxima en el método de regresión polinómica (tabla 13).

Figura 12.

Mapa de calor del LPSP para el Random Forest (método de selección de datos de entrenamiento ***aleatorio****)*



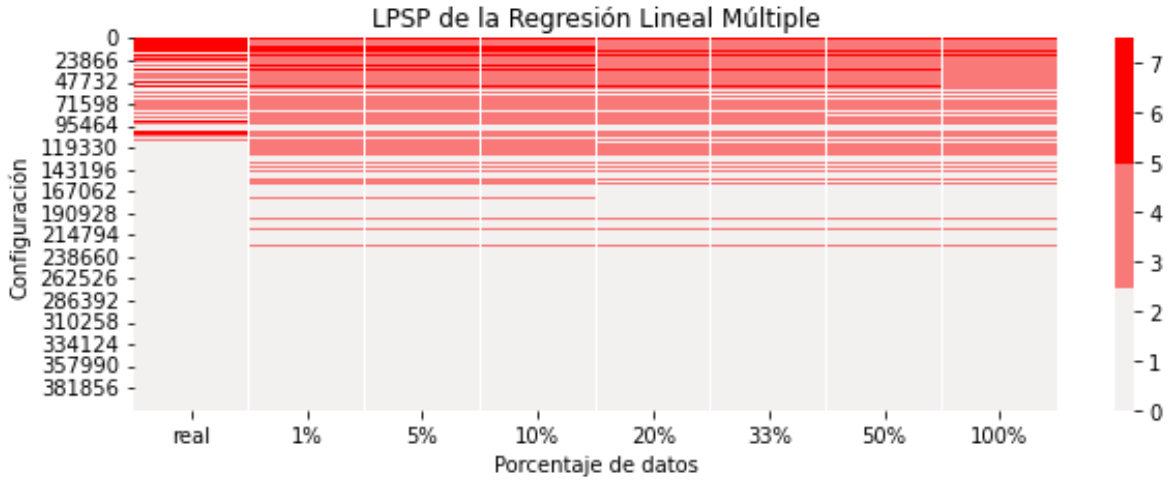
5.3.1.2. Datos Por Salto.

El comportamiento de los resultados utilizando el método de selección de datos por salto es semejante al obtenido en la selección de datos aleatorios, pero en términos de precisión (Tablas 13 y 14), el método de selección de datos aleatorios presenta mejores resultados.

Se presenta un error considerablemente grande en la regresión lineal múltiple en comparación a los otros métodos de regresión.

Figura 13.

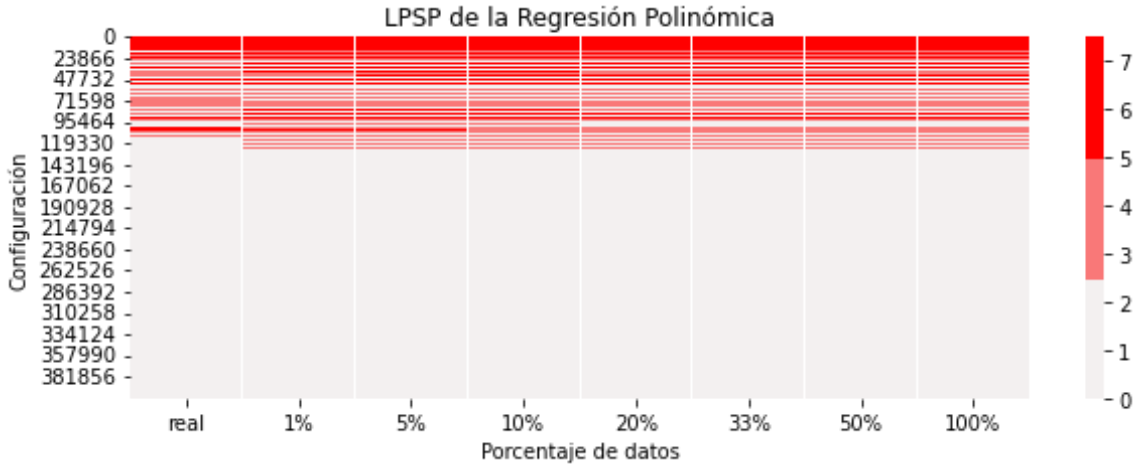
*Mapa de calor del LPSP para la regresión lineal múltiple (método de selección de datos de entrenamiento **saltos**)*



La regresión polinómica, muestra en el mapa de calor de la figura 14 que intenta acoplarse como solución al problema, pero en determinadas posiciones los resultados no son acordes al Data Set original. Al realizar la comparativa con el mapa de calor de la figura 11 utilizando el método de selección de datos aleatorios, vemos una ligera diferencia que plantea la posibilidad de que el método de selección de datos por saltos presente algunos errores dispersos en todo el Data Set resultante.

Figura 14.

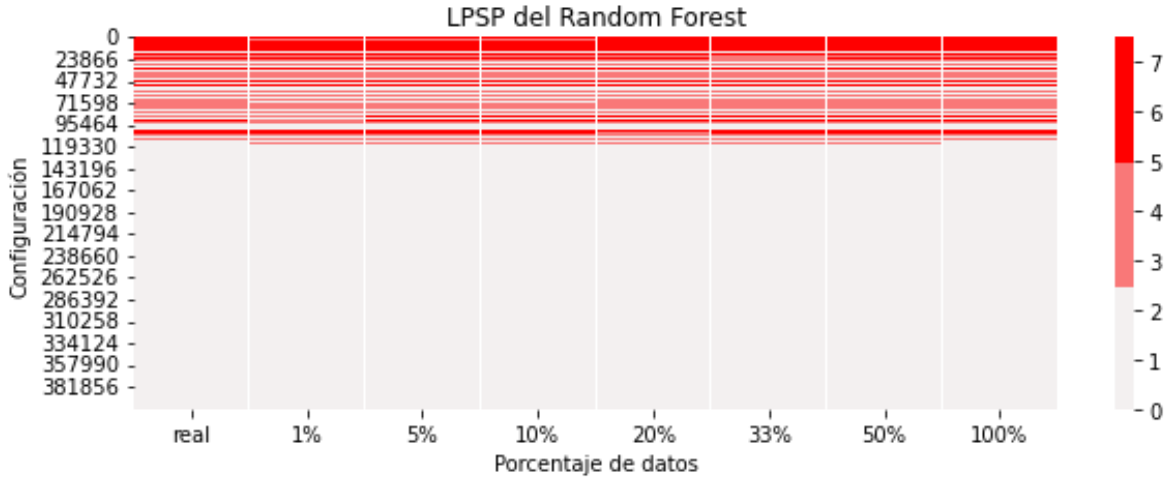
*Mapa de calor del LPSP para la regresión polinómica (método de selección de datos de entrenamiento **saltos**)*



El método de regresión Random Forest denota un comportamiento distinto en relación con los otros métodos, ya que para el mapa de calor de la figura 15 con el método de selección de datos por salto, los resultados son ligeramente mejores en términos de precisión (Tablas 13 y 14) que los obtenidos en el mapa de calor de la figura 12 con el método de selección de datos aleatorios; gráficamente se observa una gran similitud con respecto al Data Set original.

Figura 15.

*Mapa de calor del LPSP para el Random Forest (método de selección de datos de entrenamiento **saltos**)*



5.3.2. LOLH

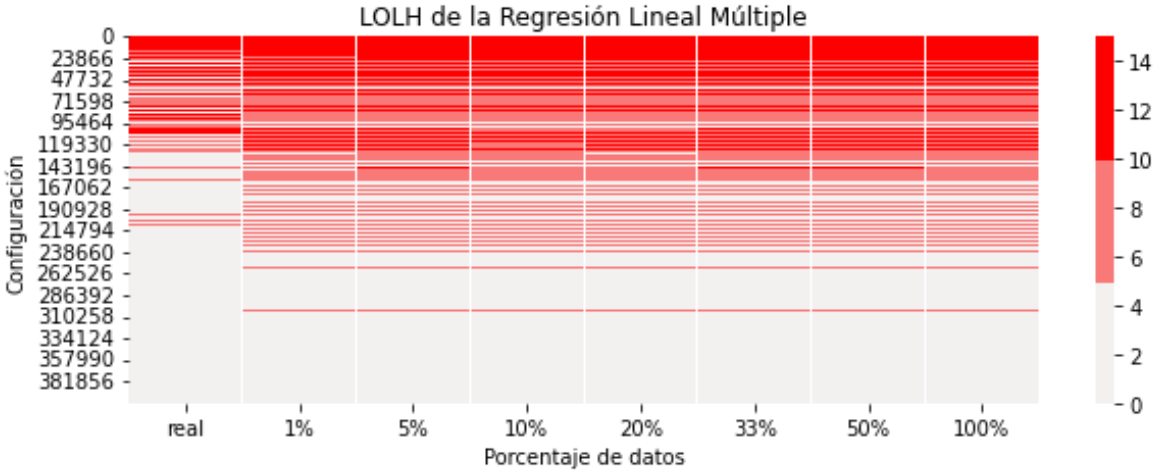
A continuación, se observan los mapas de calor para los resultados del LOLH tanto para los métodos de selección de datos aleatorios y por saltos. En la barra de la derecha, se observan los colores de la gráfica, donde el color blanco indica los valores en un rango menor o igual a 5%, el color rosado indica los valores que se encuentran entre un 5% y 10%, y el color rojo que indica los valores superiores a 10%.

5.3.2.1. Datos Aleatorios.

El indicador LOLH presenta una mayor proporción de datos superiores al límite propuesto en comparación al indicador LPSP, debido a eso, el color rojo resalta más en estas figuras. Los comportamientos de los métodos de regresión son similares a los obtenidos en el indiciador LPSP. Para este caso, el mapa de calor de la figura 16 para la regresión lineal múltiple muestra un error elevado en relación con los otros métodos de regresión.

Figura 16.

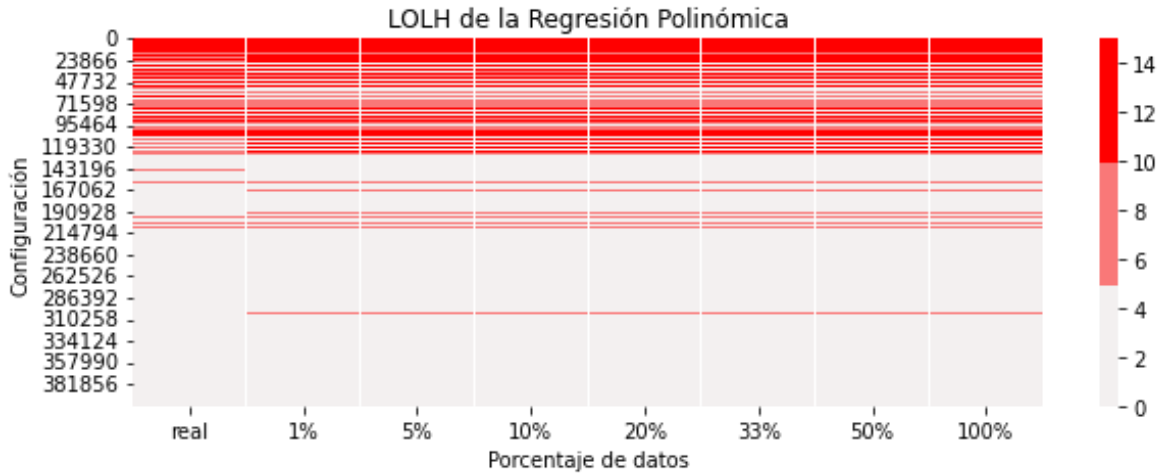
*Mapa de calor del LOLH para la regresión lineal múltiple (método de selección de datos de entrenamiento **aleatorio**)*



En el mapa de calor de la figura 17, se observa que la regresión polinómica es bastante similar al Data Set original. Presenta la misma condición que el indiciador LPSP; en algunas zonas donde las posiciones son mayores, muestran un error que no está presente en el Data Set original.

Figura 17.

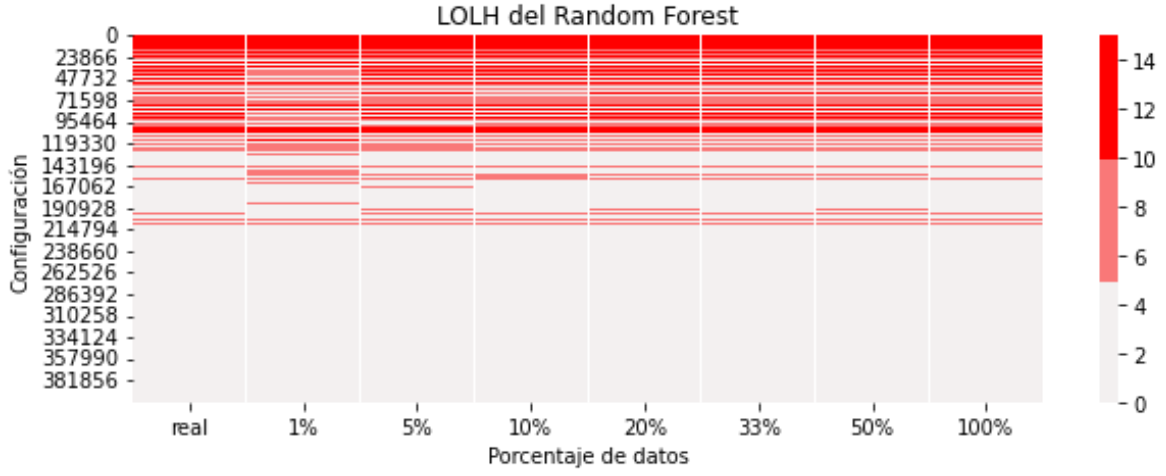
*Mapa de calor del LOLH para la regresión polinómica (método de selección de datos de entrenamiento **aleatorio**)*



Random Forest presenta nuevamente la mayor similitud con respecto al Data Set original en aquellos porcentajes de entrenamiento superiores al 10%. El 1% y 5% de los datos de entrenamiento, son los porcentajes que presentan mayor diferencia con respecto a los datos reales.

Figura 18.

*Mapa de calor del LOLH para el Random Forest (método de selección de datos de entrenamiento **aleatorio**)*

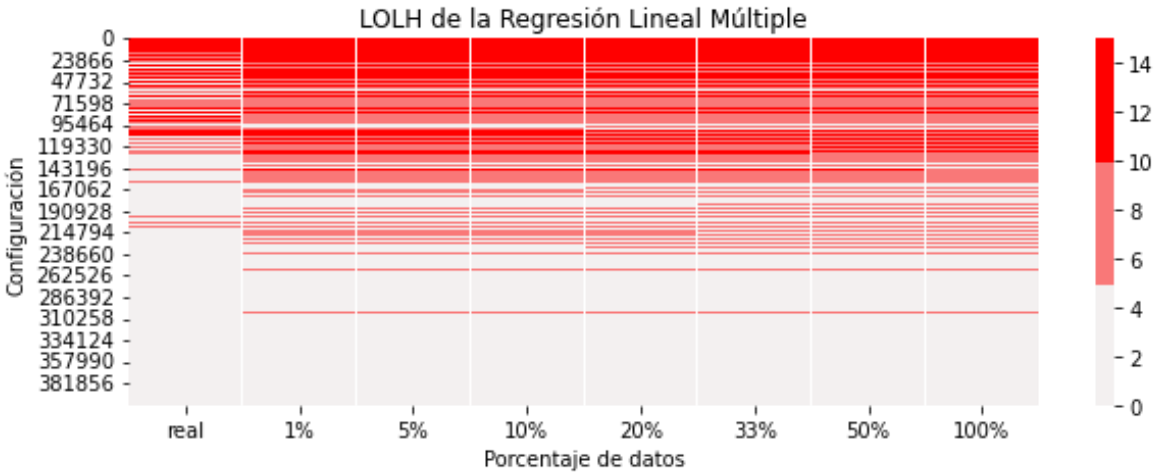


5.3.2.2. Datos Por Salto.

Para el caso de la regresión lineal múltiple con método de selección de datos por saltos, en la figura 19 se observa un color rojo bastante marcado e incluso presenta más líneas de error en posiciones superiores.

Figura 19.

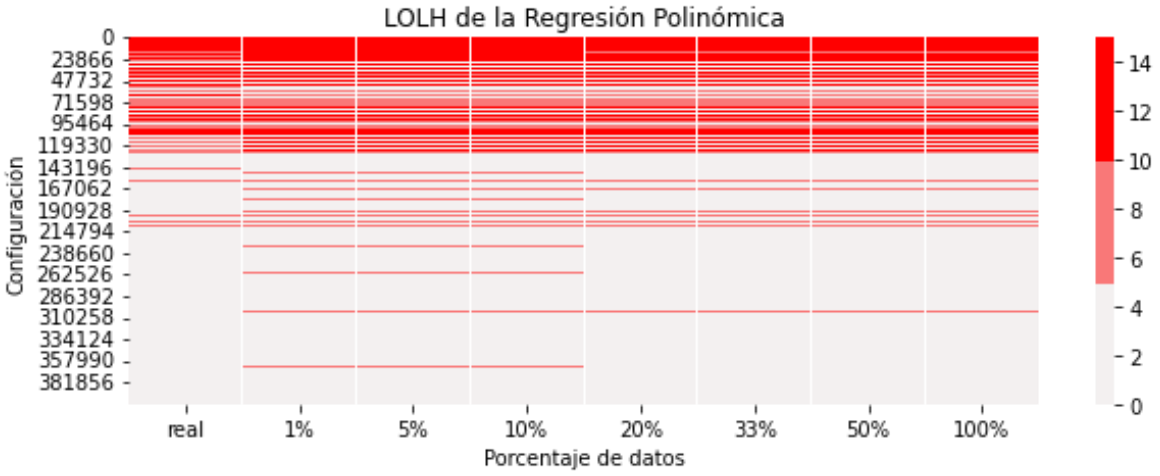
*Mapa de calor del LOLH para la regresión lineal múltiple (método de selección de datos de entrenamiento **saltos**)*



El mapa de calor de la figura 20 para regresión polinómica, presenta la misma problemática que utilizando el método de selección de datos aleatorios mostrado en la figura 17, sin embargo, solo es observable en los porcentajes de datos de entrenamiento entre el 1% y el 10%. Aun así, el resultado obtenido entre el 20% y el 100% de los datos de entrenamiento, no es visualmente igual o cercano a los valores reales.

Figura 20.

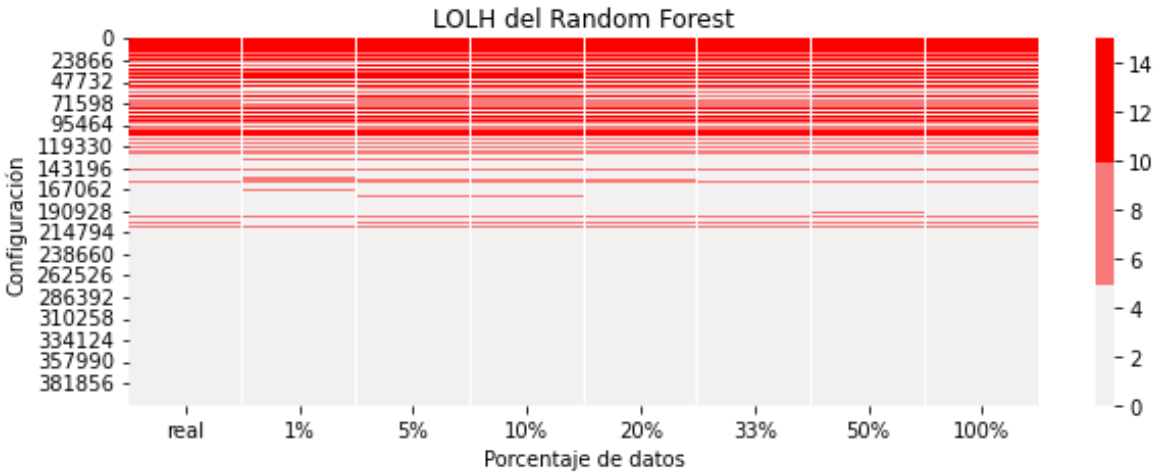
*Mapa de calor del LOLH para la regresión polinómica (método de selección de datos de entrenamiento **saltos**)*



El mapa de calor de la figura 21 para la regresión de Random Forest muestra que los resultados obtenidos son muy similares a los valores del Data Set original; este método presenta algunas discrepancias para los porcentajes de entrenamiento entre el 1% y 10%. Sigue siendo mejor gráficamente en comparación a los otros métodos de regresión.

Figura 21.

*Mapa de calor del LOLH para el Random Forest (método de selección de datos de entrenamiento **saltos**)*



5.4. **Análisis de la precisión por medio del coeficiente R^2 score**

En las siguientes figuras se observa el comportamiento de la precisión de los modelos empleados. Al escoger los datos de entrenamiento de forma aleatoria, la regresión lineal y regresión polinómica presentan una mejor precisión que por medio de saltos. En el Random Forest, se presenta una leve diferencia donde el método de selección de datos de entrenamiento por medio de saltos presenta una mejor precisión en el modelo, sin embargo, esta última técnica presenta una precisión superior al 95% sin importar el método de selección de datos para el entrenamiento. Lo anterior se complementa observando los resultados obtenidos en la sección 3.5.

Figura 22.

Gráfica de la precisión para la regresión lineal múltiple

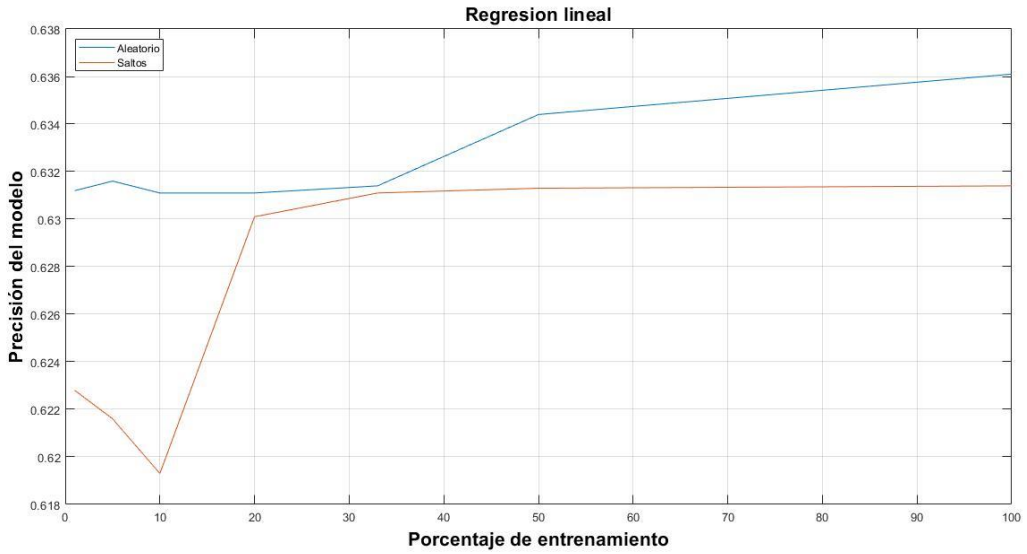


Figura 23.

Gráfica de la precisión para la regresión polinómica

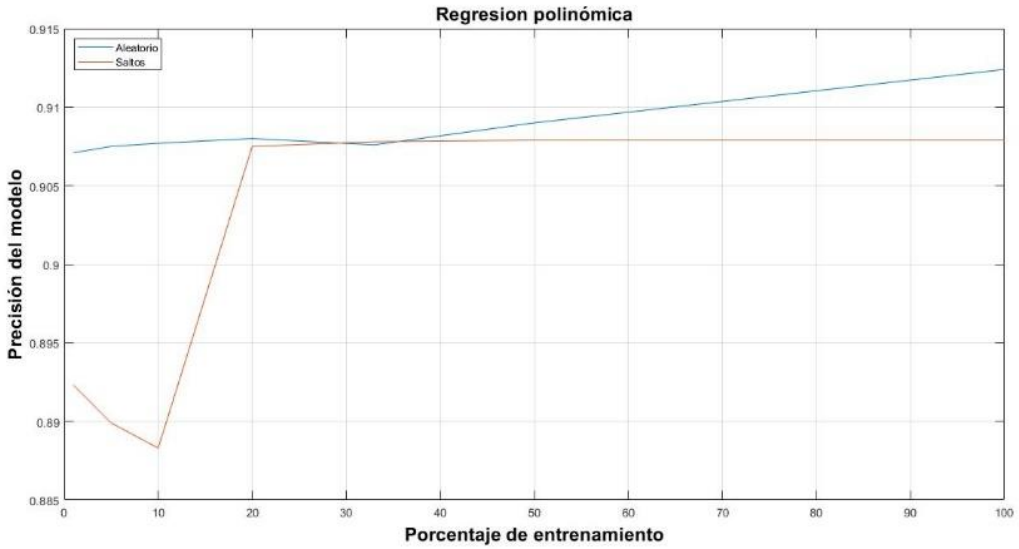
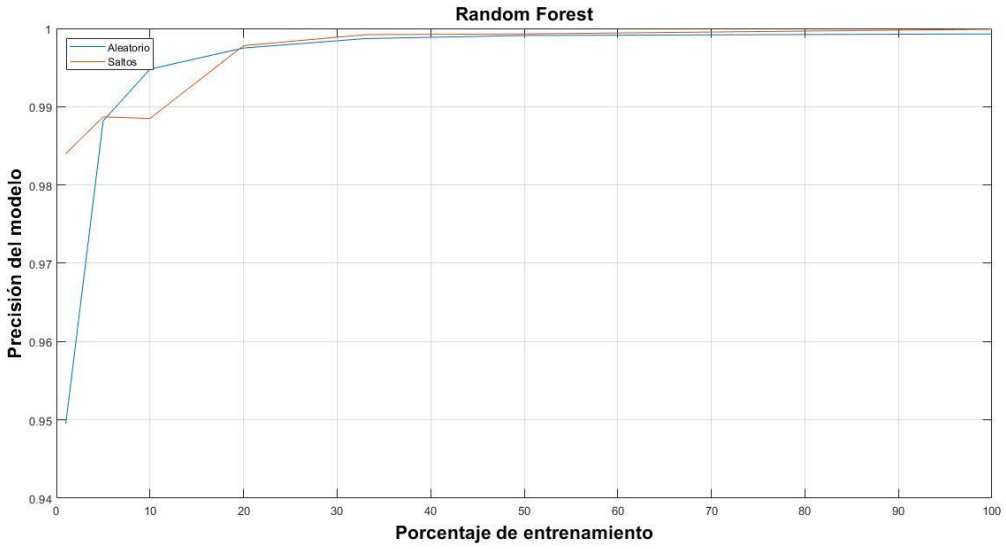


Figura 24.

Gráfica de la precisión para el Random Forest



6. Conclusiones

Con base a los resultados obtenidos, es evidente que el problema de dimensionamiento de microrredes aisladas está lejos de ser un problema lineal. Esto se sustenta por su gran diferencia con respecto a la configuración óptima y su baja precisión evaluada con el coeficiente R^2 score.

La técnica de regresión de Random Forest es la más exacta, debido a que en cada porcentaje de prueba busca acercarse lo más posible en cada uno de los parámetros de la configuración óptima. Adicionalmente esta técnica es la que presenta la mínima diferencia entre el costo de la configuración óptima del problema real y la solución obtenidas con las técnicas.

El comportamiento de regresión lineal y la regresión polinómica en los métodos de regresión es bastante similar en cada uno de los porcentajes de entrenamiento, esto quiere decir que desde el 1% hasta el 100% tienden a llevar el mismo comportamiento, a diferencia del Random Forest que en los valores obtenidos en 1% y 5% son ligeramente más errados que los valores obtenidos entre 10% y 100%. A pesar de esto, Random Forest presenta mayor semejanza con los datos reales incluso en los porcentajes más bajos en comparación a los otros métodos.

La precisión y los mapas de calor nos permiten observar que la discrepancia en los datos de la regresión lineal con respecto a los datos reales es mayor en comparación a los resultados obtenidos con el método Random Forest, gracias a esto, lo podemos posicionar como el mejor método de regresión para el problema planteado; le sigue la regresión polinómica que intenta ajustar su modelo matemático al problema, pero no logra una precisión mayor al 92%. Por último, el método de regresión lineal múltiple, que nos demuestra que este problema no puede ser representado por un modelo lineal.

Referencias Bibliográficas

Andreas C. Muller, and Sarah Guido. n.d. *Introduction to Machine Learning with Python*. Edited by O'Reilly Media. ISBN: 9781449369897.

Archdeacon, Thomas J. 1994. "Correlation and Regression Analysis: A Historian's Guide." In *ISBN 13: 9780299136543*.

Balestrieri, Michael, Salman Kahrobaee, and Peter Kim. 2021. "Application-Based Methodology for Microgrid Sizing." In *2021 IEEE Conference on Technologies for Sustainability (SusTech)*, 1–6. IEEE. https://doi.org/10.1109/SusTech51236.2021.9467468.

Bordons, Carlos, Félix García-Torres, and Luis Valverde. 2015. "Gestión Óptima de La Energía En Microrredes Con Generación Renovable." *Revista Iberoamericana de Automática e Informática Industrial RIAI* 12 (2): 117–32. https://doi.org/10.1016/j.riai.2015.03.001.

Boutros, Fouad, Moustapha Doumiati, Jean-Christophe Olivier, Imad Mougharbel, and Hadi Kanaan. 2023. "New Modelling Approach for the Optimal Sizing of an Islanded Microgrid Considering Economic and Environmental Challenges." *Energy Conversion and Management* 277 (February): 116636. https://doi.org/10.1016/j.enconman.2022.116636.

Carvajal-Romo, Gabriele, Mateo Valderrama-Mendoza, Daniella Rodríguez-Urrego, and Leonardo Rodríguez-Urrego. 2019. "Assessment of Solar and Wind Energy Potential in La Guajira, Colombia: Current Status, and Future Prospects." *Sustainable Energy Technologies and Assessments* 36 (December): 100531. https://doi.org/10.1016/j.seta.2019.100531.

Dali, Ali, Samir Abdelmalek, Abdesslem Nekkache, and Abderrezzaq Bouharchouche. 2018. "Development of a Sizing Interface for Photovoltaic-Wind Microgrid Based on PSO-LPSP Optimization Strategy." In *2018 International Conference on Wind Energy and Applications in Algeria (ICWEAA)*, 1–5. IEEE. https://doi.org/10.1109/ICWEAA.2018.8605062.

Douglas C. Montgomery, Elizabeth A. Peck, and G. Geoffrey Vining. 2021. "Introduction to Linear Regression Analysis, 6th Edition." In *ISBN: 978-1-119-57875-8*.

Iván Edgardo Jiménez Vargas. 2021. “Dimensionamiento de Microrredes Considerando Análisis de Ciclo de Vida.” Bucaramanga: Universidad Industrial de Santander.

Iván Jiménez, juan M. Rey, and German Osma-Pinto. 2020. “Sizing of Autonomous Microgrid Considering Life Cycle Emissions.” *ISBN 978-9930-541-79-1*, November.

J. Oviedo, J. Bastidas, and J. Solano. 2017. “Techniques of Analysis and Control to Improve the Stability of Electrical Microgrids: Review in the Literature ,” November.

J.D. Garzón-Hidalgo, and A.J. Saavedra-Montes. 2017. “Una Metodología de Diseño de Micro Redes Para Zonas No Interconectadas de Colombia” 20 (May).

John Hunter, Darren Dale, Eric Firing, Michael Droettboom, and the Matplotlib development. 2022. “Matplotlib Documentation.” Https://Matplotlib.Org/Stable/Index.Html. 2022.

Kamal, Md. Mustafa, Imtiaz Ashraf, and Eugen Fernandez. 2023. “Optimal Sizing of Standalone Rural Microgrid for Sustainable Electrification with Renewable Energy Resources.” *Sustainable Cities and Society* 88 (January): 104298. https://doi.org/10.1016/j.scs.2022.104298.

Kevin P. Murphy. 2012. *Machine Learning: A Probabilistic Perspective*. The MIT Press.

Khan, Faizan A., Nitai Pal, and Syed.H. Saeed. 2018. “Review of Solar Photovoltaic and Wind Hybrid Energy Systems for Sizing Strategies Optimization Techniques and Cost Analysis Methodologies.” *Renewable and Sustainable Energy Reviews* 92 (September): 937–47. https://doi.org/10.1016/j.rser.2018.04.107.

López, Andrea Ruíz, Alexandra Krumm, Lukas Schattenhofer, Thorsten Burandt, Felipe Corral Montoya, Nora Oberländer, and Pao-Yu Oei. 2020. “Solar PV Generation in Colombia - A Qualitative and Quantitative Approach to Analyze the Potential of Solar Energy Market.” *Renewable Energy* 148 (April): 1266–79. https://doi.org/10.1016/j.renene.2019.10.066.

Max Kuhn, and Kjell Johnson. 2013. *Applied Predictive Modeling*.

Ministerio de Minas y Energía, and Unidad de Planeación Minero Energética - UPME. 2015. “Integración de Las Energías Renovables No Convencionales En Colombia.” *ISBN No. 978-958-8363-26-4*.

Nelli, Fabio. 2018. *Python Data Analytics with Pandas, NumPy, and IPython*. Second Edition. Berkeley, CA: Apress. https://doi.org/10.1007/978-1-4842-3913-1.

NumFOCUS. 2022. "Pandas Documentation." Http://Pandas.Pydata.Org/Pandas-Docs/Stable/. 2022.

Rey, Juan M., Iván Jiménez-Vargas, Pedro P. Vergara, Germán Osma-Pinto, and Javier Solano. 2022. "Sizing of an Autonomous Microgrid Considering Droop Control." *International Journal of Electrical Power & Energy Systems* 136 (March): 107634. https://doi.org/10.1016/j.ijepes.2021.107634.

Rey, Juan M., Pedro P. Vergara, Javier Solano, and Gabriel Ordóñez. 2019. "Design and Optimal Sizing of Microgrids." In *Microgrids Design and Implementation*, 337–67. Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-98687-6_13.

Sandelic, Monika, Saeed Peyghami, Ariya Sangwongwanich, and Frede Blaabjerg. 2022. "Reliability Aspects in Microgrid Design and Planning: Status and Power Electronics-Induced Challenges." *Renewable and Sustainable Energy Reviews* 159 (May): 112127. https://doi.org/10.1016/j.rser.2022.112127.

Scikit-learn developers. 2022. "Documentation of Scikit-Learn." Https://Scikit-Learn.Org/0.21/Documentation.Html. 2022.

Sebastian Raschka, and Vahid Mirjalili. 2017. "Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn, and TensorFlow, 2nd Edition." In *ISBN:978-1-78712-593-3*. Packt Publishing.

The Renewables Consulting Group. 2022. "Hoja de Ruta Para El Despliegue de La Energía Eólica Costa Afuera En Colombia."

Trevor Hastie, Robert Tibshirani, and Jerome Friedman. 2009. "The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition." In *Springer Series in Statistics*.

Vanegas Chamorro, Marley, Eunice Villicaña Ortíz, and Luis Arrieta Viana. 2015. "Cuantificación y Caracterización de La Radiación Solar En El Departamento de La Guajira-Colombia

Mediante El Calculo de Transmisibilidad Atmosférica.” *Prospectiva* 13 (2): 54. https://doi.org/10.15665/rp.v13i2.487.