ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

MODELO DE PREDICCIÓN DE FALLAS EN LÍNEAS DE TRANSMISIÓN APLICADO AL SISTEMA IEEE DE 39 BARRAS USANDO REDES NEURONALES ARTIFICIALES

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO ELÉCTRICO

JORGE MOISÉS RIVERA CUZCO

DIRECTOR: MSc. PATRICIA ELIZABETH OTERO VALLADARES

CODIRECTOR: DR. ING. FABIÁN ERNESTO PÉREZ YAULI

Quito, marzo 2021

AVAL

Certifico que el presente trabajo fue desarrollado por Jorge Moisés Rivera Cuzco, bajo mi supervisión.

MSc. Patricia Elizabeth Otero Valladares
DIRECTOR DEL TRABAJO DE TITULACIÓN

Dr. Ing. Fabián Ernesto Pérez Yauli
CODIRECTOR DEL TRABAJO DE TITULACIÓN

DECLARACIÓN DE AUTORÍA

Yo, Jorge Moisés Rivera Cuzco declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentado para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

A través de la presente declaración dejo constancia de que la Escuela Politécnica Nacional podrá hacer uso del presente trabajo según los términos estipulados en la Ley, Reglamentos y Normas vigentes.

JORGE MOISÉS RIVERA CUZCO

DEDICATORIA

Dedicado a mis padres Moisés Rivera y Fanny Cuzco por su gran paciencia a lo largo de la carrera universitaria, además de ser un gran ejemplo de lucha, constancia y trabajo, siendo un apoyo incondicional en cada momento.

A Cristina Moreno la persona más especial de mi vida con quien he pasado maravillosos momentos y ha estado junto a dándome su apoyo incondicional.

A mis amigos personas valiosas con quienes he aprendido lo que es triunfar, fallar y levantarse, una segunda familia a quien se llevare en el corazón siempre.

AGRADECIMIENTO

Agradezco a mis padres por su cariño, comprensión y apoyo incondicional en todo momento a lo largo de la carrera.

Agradezco a la Msc. Patricia Otero, cuya dirección y colaboración en conjunto con el Dr. Fabián Pérez permitió el desarrollo de este trabajo.

A mis amigos Andrés, Erick, Joel, Paul, Bryan Si.,Bryan So, Santiago, Néstor, Fernanda, Alexis y demás amigos con quienes hemos compartido momentos y experiencias inolvidables a lo largo de la carrera.

ÍNDICE DE CONTENIDO

RESUMEN

En el presente trabajo se desarrolla un programa computacional mediante lenguaje de programación Python, para realizar la predicción de fallas eléctricas en las líneas de transmisión (L/T) utilizando redes neuronales artificiales (RNA) de la librería keras y Tensorflow.

Debido a la falta de datos de fallas de L/T, se inicia creando una base de datos tomando nombres y distancias de las L/T del sistema de prueba IEEE 39 barras mediante un script en lenguaje DPL del programa Power Factory. A partir de la generación de números aleatorios, se selecciona la L/T en falla, se estiman las tasas de falla, probabilidad de falla, probabilidad de reparación, distancia en la que ocurre la falla y tipo de falla. Para encontrar los tiempos de falla y reparación se usa el método de transformación inversa de las distribuciones de probabilidad. Finalmente, se obtiene la base de datos de fallas.

Con la base histórica de eventos de falla creada, su información es usada como datos de entrada a la RNA, donde la red neuronal artificial es multicapa con alimentación hacia adelante, y aprendizaje supervisado. La configuración optima de la RNA se la realiza mediante prueba y error, comparando así diferentes configuraciones (número de neuronas y número de capas ocultas) y encontrando la RNA que mejor se ajuste. La comparación entre configuraciones de RNA utiliza el cálculo del error cuadrático medio (MSE). La base de datos es dividida en sets de entrenamiento y validación, que serán utilizados en la etapa de entrenamiento y validación respectivamente.

PALABRAS CLAVE: Script, red neuronal artificial (RNA), perceptrón multicapa, líneas de transmisión, error cuadrático medio.

ABSTRACT

This work presents, a computational program using Python programming language is developed to predict electrical faults in transmission lines (L / T) using artificial neural networks (ANN) from the keras and Tensorflow library.

Due to the lack of L / T failure data, it begins by creating from a database taking names and distances of the L / T from the IEEE 39 bar test system using a DPL script from the Power Factory program. From the generation of random numbers, the L / T in failure is selected, the failure rates, probability of failure, probability of repair, distance in which the failure occurs, and type of failure are estimated. To find the failure and repair times, the inverse transformation method of probability distributions is used. Finally, the fault database is obtained.

With the historical base of failure events created, its information is used as input data to the ANN, where the artificial neural network is multilayer with forward feeding, and supervised learning. The optimal RNA configuration is done by trial and error, thus comparing different configurations (number of neurons and number of hidden layers) and finding the RNA that best fits. The comparison between RNA configurations uses the calculation of the root mean square error (MSE). The database is divided into training and validation sets, which will be used in the training and validation stage, respectively.

KEYWORDS: Script, artificial neural network, multilayer perceptron, transmission lines, mean square error.

1. INTRODUCCIÓN

Las Líneas de Transmisión en el Sistema Eléctrico de Potencia (SEP) habilitan la transferencia de energía eléctrica desde la generación hasta los centros de consumo, convirtiéndose en uno de los componentes más importantes del SEP. Dado que las L/T están expuestas a condiciones ambientales son más susceptibles a interrupciones no programadas o fallas, sin embargo, no solo las condiciones ambientales inciden en las fallas en líneas de transmisión, el aumento y disminución de carga también contribuyen en la desestabilización del sistema [1].

En los últimos años se han propuesto métodos basados en acciones de protección de relés y acciones de componentes eléctricos. Sin embargo, tienen deficiencias relacionadas con fallas en el Sistema Eléctrico de Potencia [2]. Existen eventos de falla que pueden ser repetitivos dando origen a un patrón, tales como: el crecimiento de la vegetación, incremento de descargas atmosféricas en ciertas estaciones del año. En [3] se investigan modelos para predecir las tasas de fallas en líneas de distribución de energía relacionadas con la variación de la vegetación en el tiempo. Modelos como regresión lineal, regresión exponencial, regresión lineal multivariable y una red neuronal artificial (ANN) son usados para la predicción de datos usando datos históricos, la precisión de un modelo de predicción puede ir mejorando al añadir factores o variables.

La predicción es el proceso de analizar y extraer datos históricos para predecir si hay o no una falla en el sistema eléctrico, de modo que puedan tomar medidas para prevenir accidentes y garantizar la restauración del sistema [2].

El avance de la tecnología ha permitido desarrollar herramientas y algoritmos que ayudan en el análisis, comportamiento y predicción de datos, como son: aprendizaje de maquina (Learning Machine), aprendizaje profundo (Deep Learning), redes neuronales artificiales (Artificial Neuronal Networks). Las redes neuronales artificiales se utilizan ampliamente en muchas aplicaciones de ingeniería, ya que tienen la capacidad de modelar cualquier función no lineal sin la necesidad de un modelo matemático [4]. Un ejemplo claro del uso de redes neuronales artificiales en sistemas eléctricos tenemos que en [5] se hace uso de ANN para predecir la estabilidad del sistema de energía directamente después de despejar la falla.

Puesto que el uso de ANN tiene amplias aplicaciones, el presente proyecto tiene como objetivo predecir fallas eléctricas en el sistema IEEE de 39 barras, considerando: la creación de un historial de fallas usando la simulación de Monte Carlo y tomando datos de las líneas del sistema IEEE de 39 barras, los mismo que se usarán para el entrenamiento

y validación de la red neuronal artificial. Para la predicción de las fallas eléctricas se hará uso del lenguaje de programación Python y sus librerías como: Pandas, Numpy, Sklearn, Keras y Tensorflow.

1.1 OBJETIVOS

1.1.1. OBJETIVO GENERAL

Realizar un modelo para la predicción de posibles fallas en líneas de transmisión usando el sistema de 39 barras de la IEEE basado en redes neuronales artificiales.

1.1.2. OBJETIVOS ESPECÍFICOS

Realizar investigación bibliográfica de la utilización del método de Montecarlo para simulación de eventos de fallas en sistemas eléctricos de potencia y sobre redes neuronales artificiales y su uso en predicción de fallas.

Crear la base de datos de eventos de falla a partir de la cual se trabajará para desarrollar el modelo de predicción.

Implementar el algoritmo de redes neuronales artificiales en el leguaje de Python y una interfaz para la visualización de datos y resultados.

Validar el modelo de predicción mediante la comparación de la base de datos creada y resultados obtenidos.

1.2 ALCANCE

El presente trabajo de titulación tiene la finalidad de realizar un modelo predictivo de fallas en las líneas de transmisión, mediante el uso de una red neuronal artificial para obtener las fechas y ubicaciones de posibles fallas futuras en las L/T.

Al no contar con una base de datos históricos se usará el método de Montecarlo la creación de esta en el sistema de 39 de barras de la IEEE, previamente se definirán las variables que se usar en el método de Montecarlo. De esta manera se logrará obtener una base de datos con los eventos de fallas, los cuales se convertirán en la información de entrada en el modelo de predicción en el cual se usará la metodología de redes neuronales artificiales para la predicción de fallas.

La metodología de redes neuronales artificiales se la desarrollará en el lenguaje de Python y permitirá la visualización y comparación de la base de datos generada con los resultados obtenidos.

1.3 MARCO TEÓRICO

1.3.1. LÍNEAS DE TRANSMISIÓN

La función principal de un sistema de transmisión es transferir energía eléctrica desde las plantas generadoras hacia los centros de carga.

Las plantas generadoras en conjunto con el sistema de transmisión permiten no solo el despacho económico de energía dentro de las regiones durante condiciones normales, sino también la transferencia de energía entre regiones durante emergencias como, por ejemplo: pérdida de centrales de generación eléctrica. Este último papel de la transmisión se ha vuelto particularmente importante en los últimos años debido a los desequilibrios geográficos en la disponibilidad de combustible [6].

Las líneas de transmisión se representan en función de resistencia, inductancia y capacitancia [7]. La Figura 1.1 representa el circuito equivalente de una L/T.

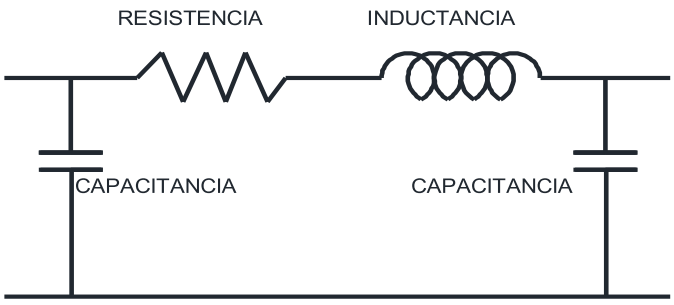


Figura 1.1. Equivalente de una línea de transmisión

1.3.2. FALLAS EN LÍNEAS DE TRANSMISIÓN

En el sistema eléctrico de potencia es estable mientras variables como, por ejemplo: voltaje y ángulo permanezcan dentro de rangos establecidos se dice que es estable o está en una condición normal, al no cumplir con estos rangos se dice que presenta una condición anormal como, por ejemplo: corto circuitos, sobre voltaje, bajo voltaje, sobrecarga, lo cual puede con llevar al colapso de todo el SEP [2].

1.3.2.1. Tipos de corto circuitos

Las fallas en el sistema de transmisión pueden ser simétricas(balanceadas) o asimétricas(desbalanceadas), las cuales tienen dos componentes. Una componente es de corriente continua la cual decrece con la constante de tiempo L/R (L: inductancia y R: resistencia) del circuito; y la segunda componente de estado estable con amplitud constante que cambia sinusoidalmente [8].

Los tipos de fallas son:

Falla Trifásica (LLL): Como se muestra en la Figura 1.2, es el contacto directo entre las tres fases y se clasifica como falla simétrica puesto que las tres fases tienen una magnitud de corriente igual, además se desfasan 120° [8].

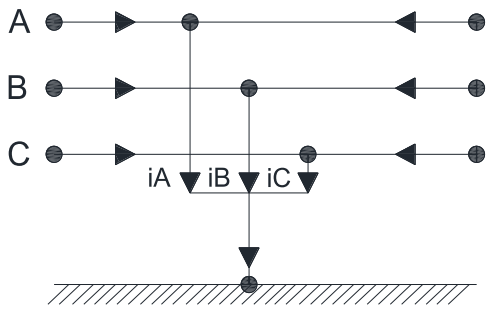


Figura 1.2. Falla Trifásica

Falla Bifásica (LL): Como se muestra en la Figura 1.3, es el contacto directo entre dos diferentes fases, las dos fases tienen una magnitud de corriente igual, y sus ángulos se desfasan 180° [8].

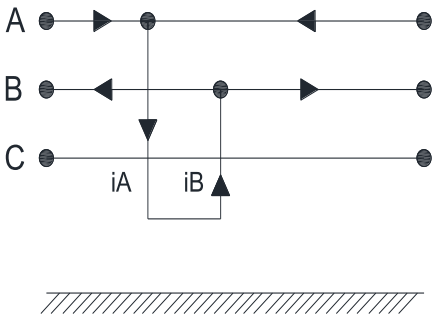


Figura 1.3. Falla Bifásica

Falla Bifásica a tierra (LLG): Como se muestra en la Figura 1.4, es el contacto directo entre dos diferentes fases y tierra, la magnitud de corriente de las dos fases es muy similar, y sus ángulos se desfasan 120° [8].

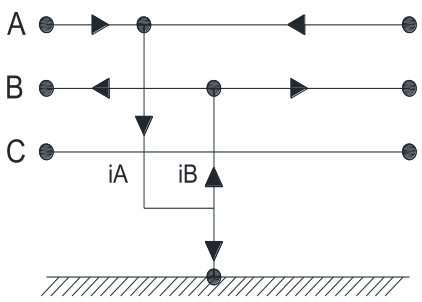


Figura 1.4. Falla Bifásica a tierra

Falla Monofásica (LG): Como se muestra en la Figura 1.5, es el contacto entre una de las diferentes fases con tierra [8].

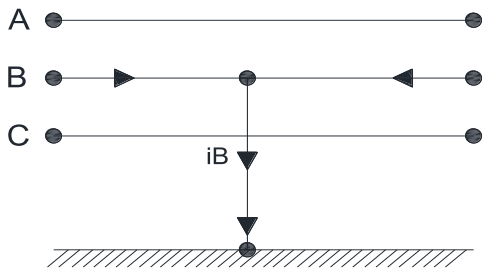


Figura 1.5. Falla Monofásica

Para el Sistema de Transmisión de Ecuador las estadísticas de ocurrencia para los tipos de falla se detallan en la ***Tabla 1.1.*** *Estadísticas de ocurrencia de fallas para el sistema de transmisión de Ecuador*Tabla 1.1.

Tabla 1.1. Estadísticas de ocurrencia de fallas para el sistema de transmisión de Ecuador [9]

Tipo de falla	Naturaleza	Ocurrencia de fallas
LLL	Balanceada	2%
LL	Desbalanceada	8%
LLG	Desbalanceada	5%
LG	Desbalanceada	85%

1.3.3. CONFIABILIDAD EN LÍNEAS DE TRANSMISIÓN

La confiabilidad se define como la probabilidad que tiene un sistema para funcionar adecuadamente durante un tiempo determinado y bajo condiciones de operación [10], en los SEP se estudia los factores que afectan la continuidad en el suministro de energía [11].

En el sistema eléctrico los componentes están sometidos a fallas cuya naturaleza es aleatoria causadas por:

1. Deterioro y envejecimiento en los materiales de los componentes eléctricos.

2. Manifestaciones del clima como las descargas atmosféricas, lluvias, viento etc.

3. Variables de operación como medio para la generación y demanda de energía.

4. Eventos al azar como accidentes de tránsito, errores de maniobras, etc.

1.3.3.1. Forma de las funciones de confiabilidad

La forma característica de muchos componentes físicos es a menudo la denominada curva de bañera y se divide en tres zonas.

ZONA I: conocida como “mortalidad infantil o eliminación de errores”, la cual se debe a errores de fabricación o diseño inadecuado. La tasa de falla disminuye en función del tiempo [12].

ZONA II: conocida como “vida útil o funcionamiento normal”, caracterizada por su tasa de falla la cual es constante. Las fallas en esta zona ocurren por casualidad y esta es la única región en la que la distribución exponencial es válida [12].

ZONA III: conocida como “desgaste o fatiga”, caracterizada por un veloz aumento de la tasa de falla [12].

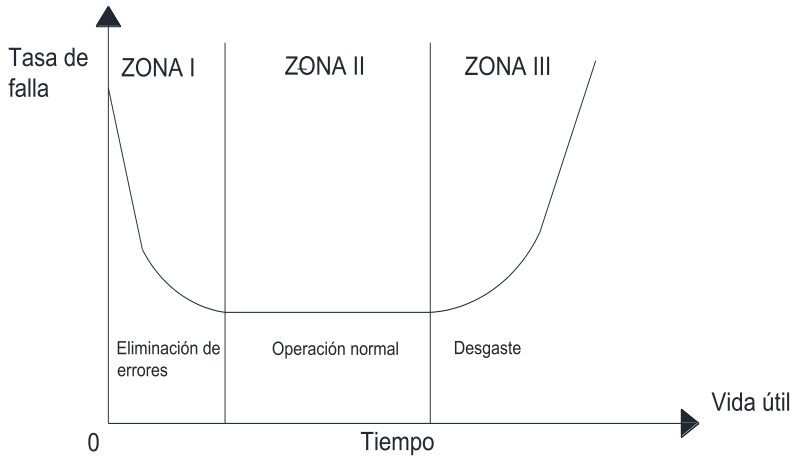


Figura 1.6. Tasa de falla durante la vida operativa de un componente

1.3.3.2. Tasa de falla(λ)

Es el número de fallas que tiene un componente en un periodo de tiempo, su unidad es las fallas por hora(f/h) o fallas por año(f/a) [13].

1.3.3.3. Tiempo medio hasta la falla (MTTF)

Es el tiempo medio para que se produzca una falla desde que el componente entro en operación por última vez luego de haber fallado [13].

1.3.3.4. Tiempo medio de reparación (MTTR)

Es el tiempo medio para que el componente en falla vuelva a operar. El MTTR no incluye los tiempos de logística, como adquisición de componentes o movilización de la tripulación [13].

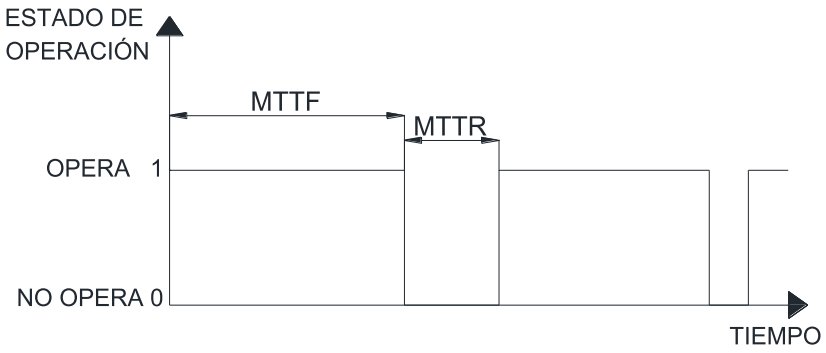


Figura 1.7. Estado de operación de componentes del SEP

1.3.4. ESTIMACIÓN DE LA TASA DE FALLA EN LÍNEAS DE TRANSMISIÓN

La secuencia de operación de los componentes de un SEP, están definidos por sus fallas o estados de operación y no operación mostrada en la Figura 1.7. No están consideradas las fallas simultaneas, fallas parciales, falla del componente mientras es reparado [14].

En los Sistemas Eléctricos de Potencia los estudios de confiabilidad requieren de modelos probabilísticos, los cuales muestran los componentes que lo conforman. Donde modelos como bloques de frecuencia y duración, distribuciones de probabilidad (Exponencial, Weibull, Binomial, etc.) y procesos estocásticos (Márkov, Poisson) son los más utilizados [14].

Los modelos de probabilísticos en el estudio de confiabilidad son construidos a partir de información sobre eventos de fallas (Número de fallas, tiempo de ocurrencia y tiempo de reparación). Sin embargo, el construir los modelos de probabilísticos presentan problemas como, por ejemplo:

- Los eventos de falla al ser aleatorios deben ocurrir para ser registrados.
- Los componentes del SEP tienen una tasa de falla son muy baja durante su vida útil.
- Se registran pocas fallas o ninguna durante largos períodos de operación.

Estimación puntual:

La tasa de fallas estimada $\hat{\lambda}$ esta defina como [14]:

$$\hat{\lambda} = \frac{n}{\sum_{i=1}^{n} ttf_i} = \frac{1}{\hat{m}} = \frac{n}{T - \sum_{i=1}^{n} ttf_r} = \frac{n}{T} \qquad (1.1)$$

Donde:

$\widehat{m}$: es la media estadística de los tiempos para la falla.

ttf_i: tiempo para la falla.

ttf_r: tiempo para la reparación.

T: periodo de tiempo.

n: número de fallas.

Los estimadores puntuales son valores promedios, donde su calidad depende del número de fallas. El en SEP la mayoría de los componentes su tiempo de reparación no se toma en cuenta respecto al tiempo de disponibilidad, lo cual permite justificar la Ecuación 1.1.

Si la variable aleatoria x tiene una muestra de información muy grande, se tendrá la seguridad de que la media estadística (estimador) será igual al valor esperado E(x), esto lo establece la Ley Fuerte de los grandes números [15].

Procesos estocásticos:

Es un conjunto de variables aleatorias definidas sobre un mismo espacio muestral o espacio de estado, donde: t es la unidad de tiempo y x(t) representa el estado del sistema [16].

1.3.4.1. Método 1: Proceso de Poisson

En un componente la tasa de falla varía durante su vida de operación tal como se muestra en la Figura 1.6. Como se mencionó anteriormente solo en la ZONA II la tasa de falla es constante, por lo cual para un componente que ha tenido pocas fallas o ninguna se dice que está en el periodo de su vida útil.

Es un proceso estocástico de Poisson homogéneo, si [11]:

- El arribo de las fallas son una a la vez.
- Cada falla es independiente.
- La tasa de llegada de fallas λ es constante.
- No se toma en cuenta el tiempo de reparación.
- La cantidad de fallas de intervalo de tiempo no afecta la cantidad de fallas durante otro intervalo de tiempo.

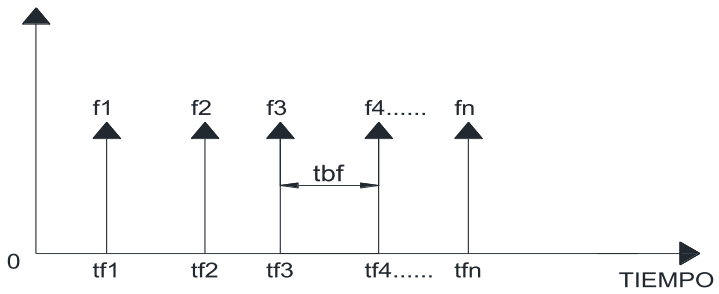


Figura 1.8. Proceso de arribo de fallas

Para el proceso de Poisson homogéneo los tiempos entre fallas (tbf) están distribuidos exponencialmente con la tasa de falla y los tiempos de llegada de fallas (tf) están distribuidos de manera uniforme [15].

La probabilidad de que ocurran n fallas está dada por [11]:

$$P[x = n] = \frac{(\lambda * t)^n}{n!} e^{-\lambda * t} \qquad (1.2)$$

Donde:

x : variable aleatoria.

λ: tasa de falla.

t: periodo de tiempo.

n: número de fallas.

La probabilidad acumulada o probabilidad que ocurran n o menos fallas está dada por [11]:

$$P[x \leq n] = \sum_{i=0}^{n} \frac{(\lambda * t)^i}{i!} e^{-\lambda * t} \qquad (1.3)$$

Donde:

x : variable aleatoria.

λ: tasa de falla.

t: periodo de tiempo.

n: número de fallas.

Usando una cantidad de n fallas para un periodo T, se determina el valor de la tasa de fallas estimada para la cual se alcanza la probabilidad de rechazo α [12].

$$\alpha = 1 - \delta = \sum_{i=0}^{n} \frac{(\hat{\lambda} * T)^i}{i!} e^{-\hat{\lambda} * T} \qquad (1.4)$$

Existe un intervalo de confianza unilateral donde se puede afirmar que $\lambda \leq \hat{\lambda}$. Siendo $\hat{\lambda}$ el límite superior con un δ % de confianza.

Tabla 1.2. Valores de $\hat{\lambda}$ [fallas/año] para $\alpha = 5\%$ usando el proceso de Poisson

n [fallas]	T [años]			
	5	10	15	20
0	0.599	0.2996	0.1997	0.1498
1	0.9488	0.4744	0.3163	0.2372
2	1.2592	0.6296	0.4197	0.3158
3	10.1	0.7754	0.5169	0.3877
4	10.1	6.2738	0.6102	0.4577

1.3.4.2. Método 2: La distribución chi-cuadrado

La dificultad de resolver la Ecuación 1.4 mediante el proceso de Poisson nos lleva aplicar la siguiente aproximación usando la distribución Chi-cuadrado [12], [14].

$$\hat{\lambda} = \frac{X^2_{\frac{\alpha}{2},v}}{2T} \qquad (1.5)$$

Donde:

v: Número de grados de libertad.

$$v = 2(n+1) \qquad (1.6)$$

$X^2_{\frac{\alpha}{2},v}$: Valor en la distribución X^2 para la cual existe la probabilidad de rechazo α con v grados de libertad.

Existe un intervalo de confianza unilateral donde se puede afirmar que $\lambda \leq \hat{\lambda}$. Siendo $\hat{\lambda}$ el límite superior con el un δ % de confianza.

Tabla 1.3. Valores de $\hat{\lambda}$ [fallas/año] para $\alpha = 5\%$ usando la aproximación a la distribución Chi- cuadrado

n [fallas]	T [años]			
	5	10	15	20
0	0.599	0.2996	0.1997	0.1498
1	0.9488	0.4744	0.3163	0.2372
2	1.2592	0.6296	0.4197	0.3158
3	1.5507	0.7754	0.5169	0.3877
4	1.8307	0.9154	0.6102	0.4577

1.3.5. PREDICCIÓN DE FALLAS ELÉCTRICAS

A medida que las redes eléctricas evolucionan (se expanden y aumentan las cargas) la confiablidad y estabilidad del SEP se ve afectado. El principal objetivo de la predicción de fallas es aumentar la confiabilidad y la estabilidad, a su vez puede prevenir pérdidas económicas en el transporte de energía eléctrica.

La predicción de fallas es el proceso de analizar y extraer datos históricos para pronosticar si existe o no una falla en el SEP, de modo que se puedan tomar medidas para prevenir accidentes y la restauración del sistema. En general, hacer un buen uso de los datos históricos puede mejorar el rendimiento de la predicción de fallas y garantizar la confiabilidad y estabilidad de un sistema eléctrico. La investigación sobre metodologías de predicción de fallas basados en redes neuronales artificiales (ANN) y en datos históricos apenas ha comenzado a aparecer en los últimos años [2].

1.3.6. MÉTODOS DE PREDICCIÓN DE FALLAS ELÉCTRICAS

Los métodos de predicción están divididos en: cualitativos y cuantitativos [17].

Método cualitativo: Este método se caracteriza por que el pasado no aporta información relacionada con el proceso que se está considerando. En este método las estadísticas toman un papel secundario [17].

Método cuantitativo: Se parte de la idea de que existe información almacenada sobre el proceso que se va a estudiar. Este tipo de información aparece en forma de series temporales [17]. Este método se divide a su vez en métodos: estadísticos y de Inteligencia artificial [18], para los cuales se detallará su clasificación a en la siguiente sección.

1.3.6.1. Método estadístico

Los métodos estadísticos usan el principio de matemáticas o estadística, son adecuados para datos lineales.

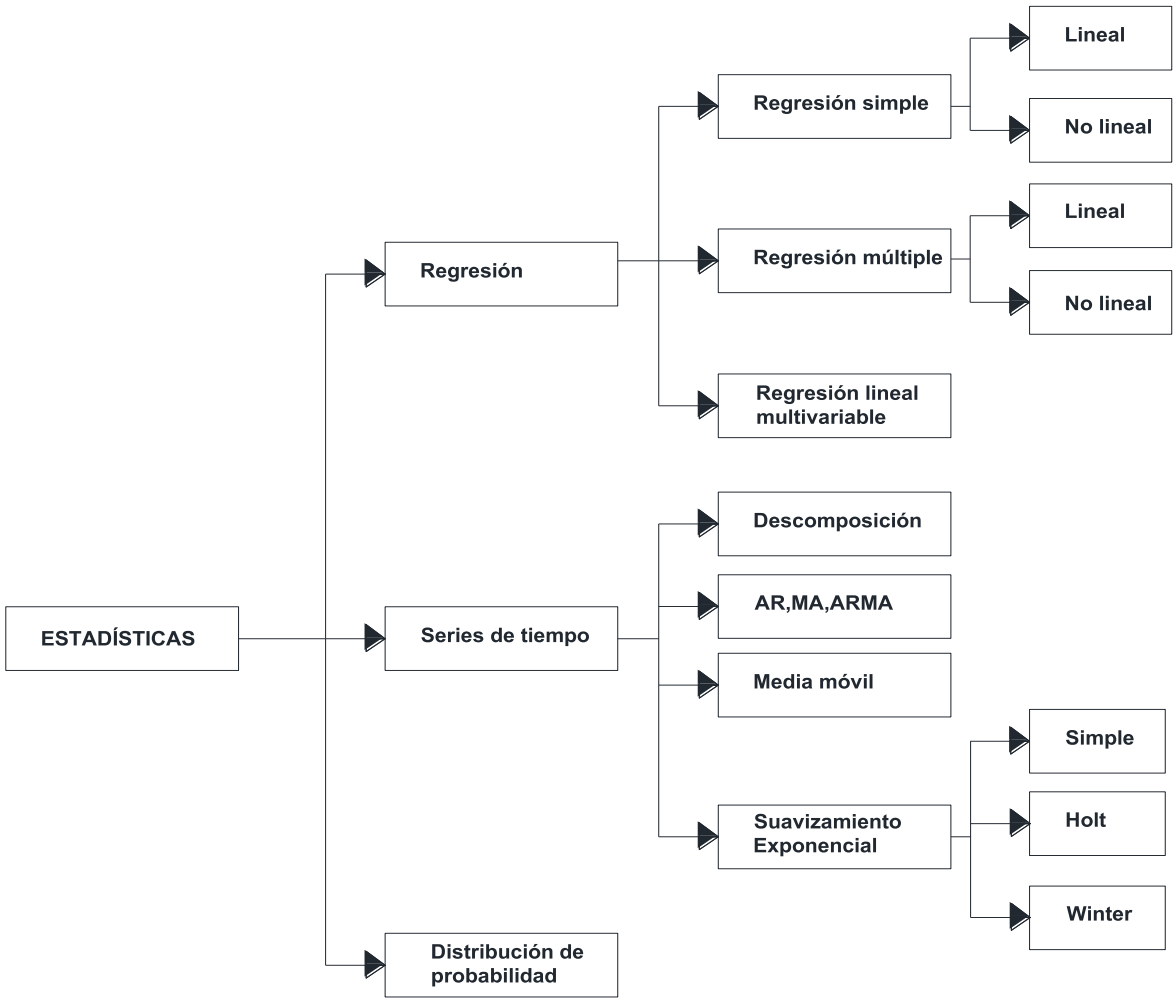


Figura 1.9. Clasificación de los métodos estadísticos

1.3.6.2. Método de inteligencia artificial

La inteligencia artificial es una herramienta capaz de resolver problemas de análisis de datos históricos, encontrado tendencias, patrones y pronósticos, en una gran cantidad de datos [19].

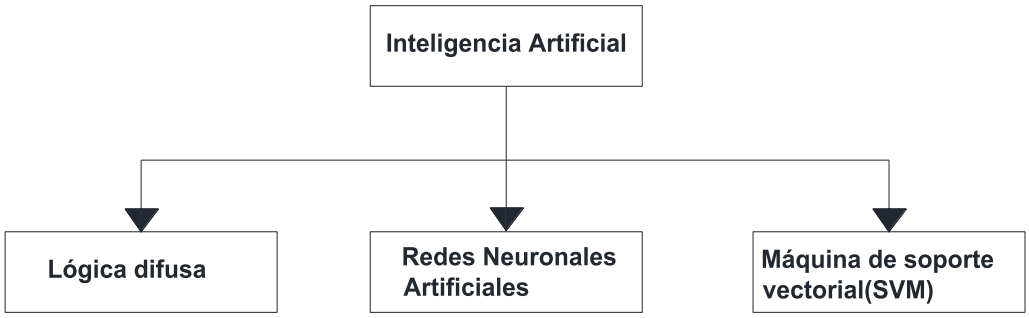


Figura 1.10. Clasificación de los métodos de Inteligencia artificial

Los modelos de inteligencia artificial para predicciones son algoritmos que modelan el comportamiento de las variables en sistemas más complejos los cuales tiene comportamientos no lineales.

1.3.6.3. Métodos híbridos

Estos métodos resultan de la combinación de dos o más técnicas de predicción, como son las técnicas estadísticas y de inteligencia artificial. La finalidad es lograr una mayor precisión, menor tiempo de procesamiento y menor error en el entrenamiento.

Combinaciones de métodos [2], [19], [20]:

- Series temporales y redes neuronales.
- Regresión múltiple y algoritmos genéticos.
- Redes neuronales y algoritmos genéticos.
- Lógica difusa y redes neuronales.
- Redes neuronales y SMV.

1.3.7. PRE-PROCESAMIENTO DE DATOS

1.3.7.1. PREPROCESAMIENTO Y POST PROCESAMIENTO

Muchas de las veces es ventajoso aplicar transformaciones de procesamiento previo a la información de entrada antes de que se usen en la red neuronal. De manera similar, la salida o salidas de la red neuronal se procesan para dar los valores de salida requeridos [21].

Los pasos de preprocesamiento y post procesamiento consisten en transformaciones simples o pueden involucrar procesos adaptativos. El preprocesamiento suele ser una de las fases más fundamentales en el desarrollo de una solución ya que tiene un efecto significativo en el rendimiento [21].

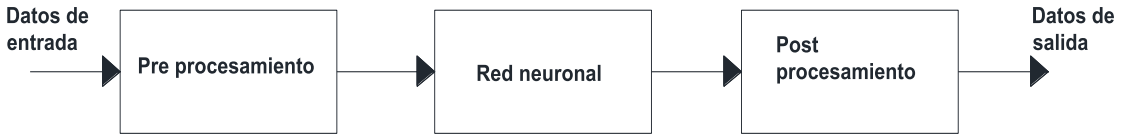


Figura 1.11. Preprocesamiento y post procesamiento de datos

1.3.7.2. Normalización de datos

Una de las formas más comunes de preprocesamiento consiste en un cambio de escala lineal de las variables de entrada [4]. La normalización de datos es una herramienta de cambio de escala, los datos son reducidos a rangos entre [0,1] o [-1,1] [22]. Al realizar esta normalización se puede hacer que todas las entradas tengan valores similares [4].

$$x_{normalizado} = \frac{(x - x_{\min})(d_2 - d_1)}{x_{\max} - x_{\min}} + d_1 \tag{1.7}$$

Donde:

x: Valor a normalizar.

$x_{\min}$, $x_{\max}$: Rango de valores mínimos y máximos para los datos x.

d_2, d_1: Rango de valores a ser reducido.

1.3.7.3. Datos faltantes

Un problema común es que algunos de los valores de entrada pueden faltar en la base de datos [21]. Los datos faltantes pueden deberse a: forma manual de tomar los datos, error de mediciones, error en los equipos, etc [23].

Los valores faltantes en una base de datos pueden ser tratados de dos formas:

1. Eliminación de casos: Es el método más usado, pero a su vez tiene una gran desventaja, puesto que, se pierde información y se trata con una muestra menor [24].
2. Imputación de un valor estimado: Este método sustituye el valor faltante por el promedio de los datos disponibles [24]. También se lo puede sustituir por la mediana calculada con los datos disponibles, o sustituir por un valor calculado por una regresión basado en los valores completos de la base de datos histórica [23].

1.3.7.4. Series de tiempo

Muchas aplicaciones de las redes neuronales involucran datos que cambian en función del tiempo. El objetivo en la gran mayoría es pronosticar el valor de una variable en un corto periodo de tiempo en el futuro [4]. La serie de tiempo se define como una secuencia de observaciones ordenadas [25].

El análisis de series de tiempo lleva a características como: comportamientos cíclicos, tendencias creciente o decreciente, estacionalidad o no estacionalidad [25].

Existen dos métodos para establecer los valores futuros de la variable a pronosticar, los cuales son [26]:

Método de predicción univariante: Conocido como método de series temporales, establecen la relación entre los valores pasados de la variable a predecir. Matemáticamente el valor en un futuro inmediato de la variable a predecir es una función de sus valores pasados como se indica en la Ecuación 1.8 [26].

$$x_{t+1} = f(x_t, x_{t-1} \dots \dots . x_{t-n)} \tag{1.8}$$

Donde:

x_{t+1}: Valor a predecir.

x_t, x_{t-n}: Valores del pasado.

Método de predicción multivariante: Conocido como método causal, establecen la relación entre los valores pasados de la variable a predecir y también de los valores pasados de variables explicativas. Matemáticamente el valor en un futuro inmediato de la variable a predecir es una función de sus valores pasados como se indica en la Ecuación 1.9 [26].

$$x_{t+1} = f(x_t, x_{t-1} \dots x_{t-n}, u_t, u_{t-1} \dots . u_{t-n}, v_t, v_{t-1} \dots v_{t-n}) \tag{1.9}$$

Donde:

x_{t+1}: Valor a predecir.

$x_t, x_{t-n}, u_t, u_{t-n}, v_t, v_{t-n}$: Valores del pasado.

1.3.8. REDES NEURONALES ARTIFICIALES

Las redes neuronales artificiales (RNA) son sistemas inspirados en los sistemas neuronales biológicos. Las RNA tienen un conjunto de elementos simples de

procesamiento, conocidos como nodos o neuronas, enlazados entre sí [26]. Las RNA son capaces de procesar información para obtener una salida inteligente.

Las RNA se pueden visualizar como una serie de capas conectadas que forman una red que enlaza los valores de características de una observación en un extremo y el valor objetivo (por ejemplo, la clase de observación) en el otro extremo [27]. Tienen la habilidad de adaptar su funcionamiento a diferentes ambientes al cambiar las conexiones entre neuronas, lo que significa que pueden aprender de la experiencia [28].

Ventajas y desventajas de las redes neuronales artificiales [29]:

Ventajas:

- Las RNA pueden analizar algoritmos por medio de un proceso de aprendizaje.
- No se necesita tener detalles matemáticos para usar las RNA.
- Da solución a los problemas no lineales.
- Gracias a la robustez de una RNA esta continuará trabajando si fallan elementos de procesamiento.

Desventajas:

- Para cada problema la RNA es entrenada. Además, se debe realizar varias pruebas para determinar la arquitectura optima. El entrenamiento puede ser largo y podría requerir varias horas de la computadora (CPU).
- Las RNA necesita de una gran cantidad de información ya que las redes son entrenadas en lugar de ser programadas.

Las dendritas, cuerpo o soma y axón son los tres principales componentes que forman una neurona como se muestra en la Figura 1.12.

Las redes neuronales artificiales están dadas por capas y se las identifica de la siguiente manera:

- **Capa de entrada:** Es la capa en la que ingresa la información o señales desde el exterior [30].
- **Capas de oculta:** Las capas ocultas se ubica dentro del sistema, no tienen contacto con el exterior [30].
- **Capa de salida:** Es la capa que envía la información o señales resultantes hacia el exterior [30].

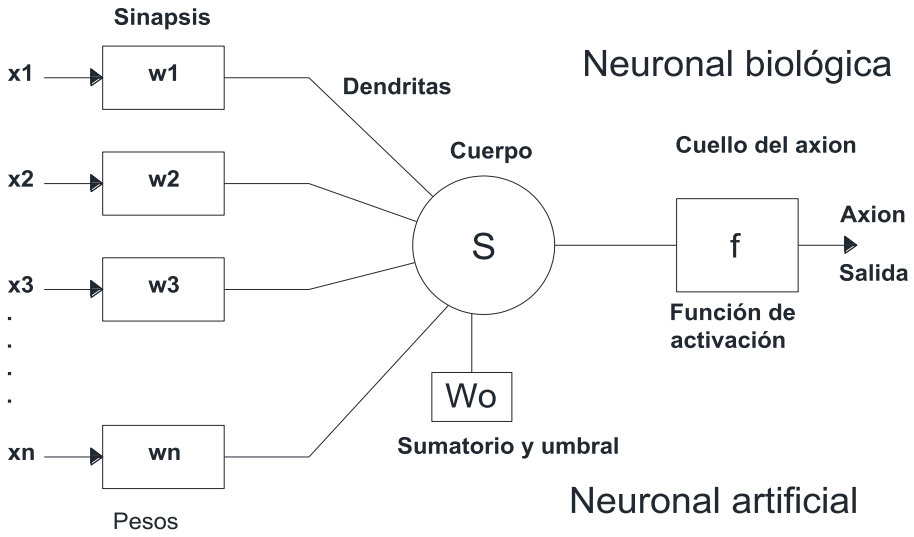
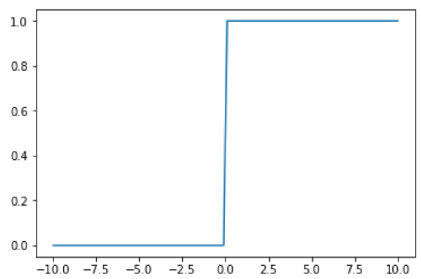


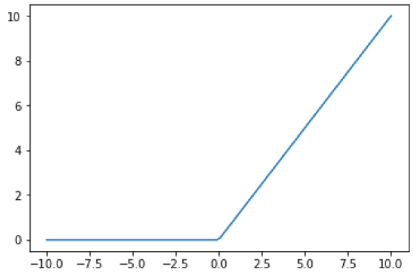
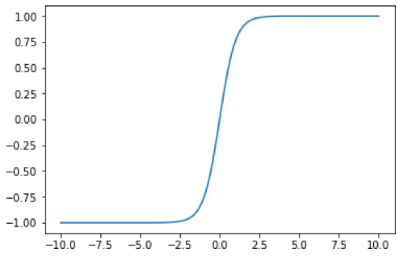
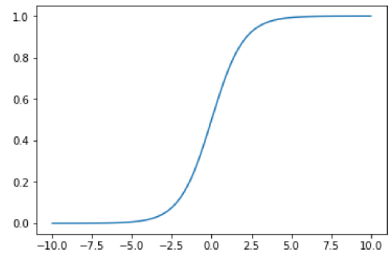
Figura 1.12. Componentes principales de una red neuronal biológica y artificial [31]

En una RNA a la conexión entre neuronas se asocia un peso. Estos pesos son los encargados que la red neuronal adquiera conocimiento. El valor de los pesos pueden ser positivos, negativos e incluyo de cero. Si tenemos dos neuronas A y B, al tomar el peso un valor positivo indica interacción entre las neuronas A y B y se lo denomina excitador. Un valor negativo indica que la neurona A ha sido activada y mandará |una señal a la neurona B que tenderá a desactivar a ésta y se la denomina inhibidora. Por último, el valor de cero indica que no hay conexión entre las neuronas A y B [30]. Los valores de los pesos de una RNA siguen la regla de propagación, la cual modifica estos valores en base a la interacción con el medio y resultado de experiencias [32].

Una RNA sigue una regla o función de activación, la cual determina la activación o el estado de una neurona, las funciones de activación pueden ser lineales o no lineales [33]. La Tabla 1.4 resume las funciones de activación que suelen usarse.

Tabla 1.4. Funciones de activación

Función	Ecuación	Gráfico
Escalón	$y(x) = 1, \quad si\ x \geq 0$ $y(x) = 0, \quad si\ x < 0$	

Función	Ecuación	Gráfico
Rampa o ReLU	$y(x) = x, \quad si\ x \geq 0$ $y(x) = 0, \quad si\ x < 0$	
Tangente hiperbólica	$y(x) = \tanh(x)$	
Sigmoidal	$y(x) = \frac{1}{1 + e^{-x}}$	

Donde:

$y(x)$: Variable dependiente.

x : Variable independiente.

La salida de una RNA existe una regla o función de salida, la cual transforma el estado de activación en una señal de salida [32].

1.3.9. CLASIFICACIÓN DE REDES NEURONALES ARTIFICIALES: SEGÚN LA TOPOLOGÍA

1.3.9.1. Red neuronal monocapa o Perceptrón simple

Es la red neuronal más básica, está conformada por una capa de neuronas donde se proyectan las entradas a una capa de neuronas de salida en la cual se realizan los diferentes cálculos [34].

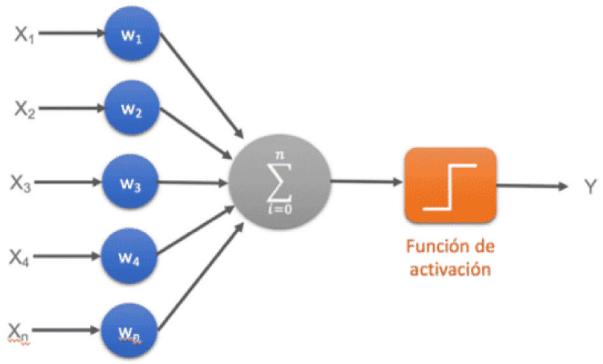


Figura 1.13. Red neuronal monocapa [34]

1.3.9.2. Red neuronal multicapa o Perceptrón multicapa

La RNA multicapa está compuesta de capas intermedias llamadas capas ocultas, ubicadas entre las capas de entrada y salida. La red puede estar total o parcialmente conectada, esto dependerá del número de conexiones que tenga la red [34].

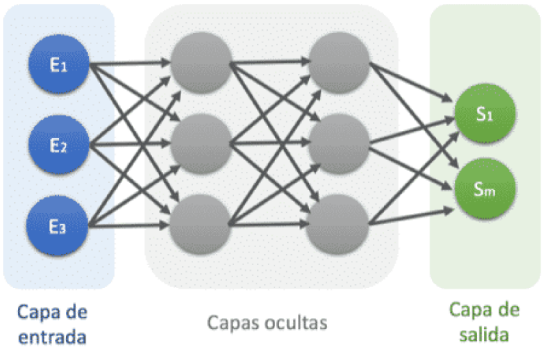


Figura 1.14. Red neuronal multicapa [34]

1.3.9.3. Red neuronal Convolucional (CNN)

La red neuronal convolucional se diferencia de la red multicapa ya que las neuronas no se conectan con todas y cada una de las posteriores capas, dando como resultado la reducción del número de neuronas y la complejidad computacional que necesita para su ejecución [34].

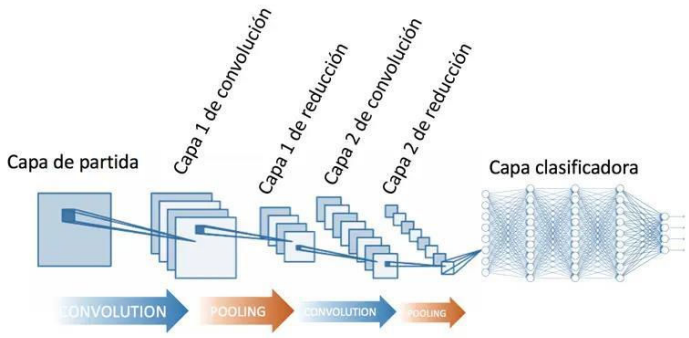


Figura 1.15. Red neuronal convolucional [34]

Las CNN tiene características de las redes multicapa para la predicción de series de tiempo, es decir, tiene entradas multivariantes, salidas multivariante y aprendizaje de relaciones funcionales arbitrarias pero complejas, no requieren que el modelo aprenda de manera directa de las observaciones de retraso. Por otra parte, el modelo logra aprender una representación de una secuencia de entrada grande que sobresale para el problema de pronosticar [35].

1.3.9.4. Red neuronal Recurrente (RNN)

Las redes neuronales recurrentes se caracterizan por permitir conexiones arbitrarias entre las neuronas, además puede crear ciclos en las neuronas, con esto se logra crear la temporalidad, dando como resultado una red con memoria [34].

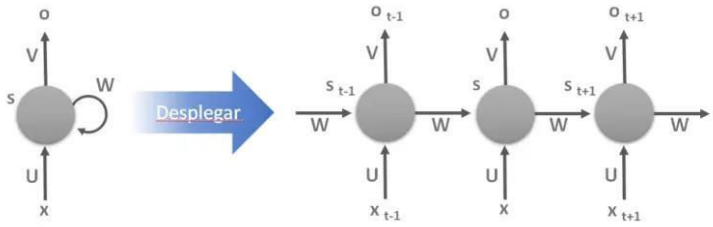


Figura 1.16. Red neuronal recurrente [34]

La información ingresada en el momento t en la entrada, son transformados y circulan por la red aun en los instantes de tiempo posteriores t + 1, t + 2, …

1.3.10. CLASIFICACIÓN DE REDES NEURONALES ARTIFICIALES: SEGÚN EL MÉTODO DE APRENDIZAJE

1.3.10.1. Aprendizaje supervisado

Es realizado con un entrenamiento controlado por un supervisor que determina la respuesta que se debe generar para cada entrada. Al contar con un supervisor este controla la salida y de no ser la correcta, cambia los pesos de las conexiones, con la finalidad de que la salida obtenida se ajuste a la deseada [34].

1.3.10.2. Aprendizaje no supervisado

No requiere influencia externa para ajustar los pesos. Este tipo de aprendizaje pretende encontrar las características, regularidades, correlaciones o categorías que se puedan establecer entre los datos que se presenten como entrada. La interpretación de sus datos depende de su estructura y del algoritmo de aprendizaje empleado [34].

1.3.10.3. Aprendizaje por refuerzo

Es considerado un aprendizaje lento incluso mayor al aprendizaje por corrección de errores, para este aprendizaje no se requiere de un conjunto completo de información exacta de salida, solo se indica si la información es aceptada o no, con esto el algoritmo logra ajustar los pesos basado en un mecanismo de probabilidades [34].

1.3.11. SIMULACIÓN DE MONTECARLO

En los sistemas eléctricos el Método de Montecarlo es aplicado generalmente en los sectores de generación y transmisión de energía.

Este método necesita la configuración de las tasas de falla y de reparación de cada uno de los componentes del sistema.

El método de Montecarlo consiste en simular situaciones en gran cantidad, generadas de manera aleatoria o pseudoaleatoria, donde los valores de los indicadores de confiabilidad pertenecen a los momentos de las distribuciones de probabilidad asociadas al estado del sistema.

Una aplicación de simulación puede involucrar varios de los tipos de simulación mostrados en la Tabla 1.5.

Tabla 1.5. Clasificación de los tipos de simulación [16]

1	Estática o dinámica	Estática	Representa el sistema o proceso en estudio para un tiempo particular o es una representación en la cual el tiempo no desempeña ningún papel.
		Dinámica	Es una representación de un sistema o proceso que evoluciona con el tiempo. Generalmente, se expresa el modelo matemático por medio de ecuaciones diferenciales.
2	Determinística o estocástica	Determinística	Las variables aleatorias no están consideradas en el modelo.
		Estocástica	El modelo considera variables aleatorias.
3	Discreta o continua	Discreta	Las variables son discretas en el modelo.
		Continua	Las variables son continuas en el modelo
4	Secuencial o no secuencia	Secuencial	Los números aleatorios que se generan conforman secuencias que definen otras variables.
		No secuencial	Los números aleatorios generados no conforman una secuencia para definir otras variables.
5	Secuencial sincrónica	Sincrónica	El paso del reloj en la simulación es fijo, por ejemplo, cada hora, días, semana, etc.
	Secuencial asincrónica	Asincrónica	El paso del reloj de la simulación es variable y se determina según un evento, por ejemplo, cada que falla un componente.

La simulación de Montecarlo más usada es el estático no secuencial.

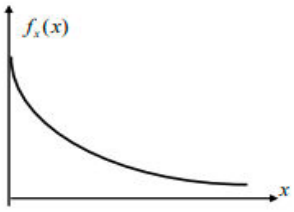
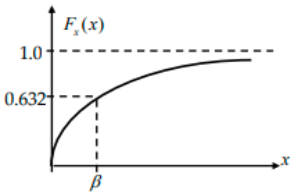
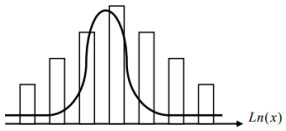
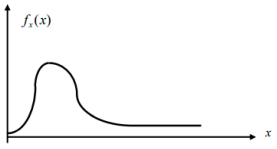
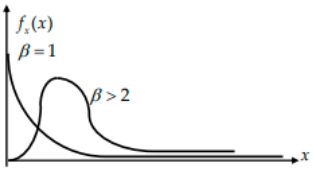
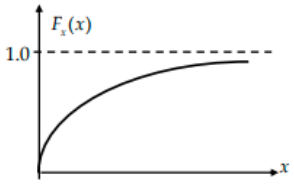
El método de Montecarlo presenta las siguientes ventajas [36]:

- Permite usar diferentes distribuciones en el modelamiento de tiempos para salidas y restauraciones de los componentes.
- Facilita la resolución de sistemas en donde no se cuenta con soluciones analíticas. Como, por ejemplo, sistemas donde uno o algunos de los componentes que lo conforman usan el modelo de la distribución Gaussiana para el tiempo para salida o restauración.
- Facilita la obtención de distribuciones de probabilidad para los índices de confiabilidad de los puntos de carga.
- No es necesario realizar cambios en el software, los cambios son realizados en la base de datos.

1.3.12. DISTRIBUCIONES DE PROBABILIDAD

Una distribución de probabilidad es un modelo matemático que se usa para representar fenómenos aleatorios [16].

Tabla 1.6. Distribuciones de probabilidad [16]

Distribución de probabilidad	Función de densidad de distribución de probabilidad f(x)	Función de distribución de probabilidad F(x)
Exponencial $f(x) = \lambda e^{-\lambda x}$ $F(x) = 1 - e^{-\lambda x}$		
Log normal $f(x) = \frac{1}{x\sqrt{2\pi}\sigma} e^{\frac{(\ln(x)-\mu)^2}{2\sigma^2}}$	 	No contiene forma analítica
Weibull $f(x) = \alpha * \beta * x^{\beta-1} e^{-\lambda x}$ $F(x) = 1 - e^{-\alpha x^{\beta}}$		

1.3.13. MÉTRICAS PARA EVALUAR EL DESEMPEÑO DE UN MODELO

1.3.13.1. Error residual

El error residual se da con la diferencia entre valores pronosticados del modelo y valores reales durante el período de evaluación [37].

$$error = y - \hat{y} \tag{1.10}$$

Donde:

y : Valor real de la variable.

$\hat{y}$: Valor estimado de la variable.

1.3.13.2. Error cuadrático medio (MSE- Mean Square Error)

El MSE es la métrica más común para evaluar el desempeño de pronóstico de un modelo, donde cada error es elevado al cuadrado y después se calcula su media. Mientras más bajo sea el valor del MSE mejor será el pronóstico o predicción del modelo [37].

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{1.11}$$

Donde:

y : Valor real de la variable.

$\hat{y}$: Valor estimado de la variable.

n : Numero de datos usados.

1.3.13.3. Error cuadrático medio (RMSE- Root Mean Square Error)

En el análisis estadístico de datos, se usa la desviación típica en lugar de la varianza, ya que la desviación es medida en unidades iguales a la variable que se está analizando, lo que facilita la interpretación de los resultados del análisis. Análogamente, se puede calcular la raíz cuadrada del MSE obteniendo una medida de error denominada RMSE [37], [38].

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{1.12}$$

Donde:

y : Valor real de la variable.

$\hat{y}$: Valor estimado de la variable.

n : Numero de datos usados.

1.3.13.4. Error absoluto medio (MAE- Mean Absolute Error)

Al utilizar los errores al cuadrado hace que las medidas sean sensibles a la presencia de anomalías o datos atípicos. El efecto que produce cada error individual es proporcional al cuadrado del tamaño del error, por lo que errores más grandes tienen efectos desproporcionados sobre medidas de error como el MSE o RMSE [38]. Por lo tanto, el MAE no penaliza los errores grandes tanto como el MSE [37] además, es robusto frente a la presencia de ruido y valores atípicos del conjunto de datos [38].

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{1.13}$$

Donde:

y : Valor real de la variable.

$\hat{y}$: Valor estimado de la variable.

n : Número de datos usados.

1.3.13.5. Error absoluto medio porcentual (MAPE- Mean Absolute Percentage Error)

Mide el error porcentual, dando una idea del tamaño de los errores de los valores pronosticados del modelo y los valores reales durante el período de evaluación [37], [38].

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \hat{y}_i|}{|y_i|} \quad (1.14)$$

Donde:

y : Valor real de la variable.

$\hat{y}$: Valor estimado de la variable.

n : Número de datos usados.

1.3.14. PYTHON: LENGUAJE DE PROGRAMACIÓN

1.3.14.1. Biblioteca Numpy

Es una biblioteca de Python que facilita un objeto de matriz multidimensional y una gran diversidad de rutinas para operaciones matrices de manera rápida, incluidas matemáticas, lógicas, clasificación, transformadas discretas de Fourier, simulación aleatoria, operaciones estadísticas que sean básicas y más [39].

1.3.14.2. Biblioteca Pandas

Es una biblioteca de Python que brinda estructuras de datos rápidas, flexibles y expresivas diseñadas para que al momento de trabajar con datos "relacionales" o "etiquetados" sea mucho más fácil. Con un nivel alto para efectuar análisis de datos reales y prácticos en Python [40].

1.3.14.3. Biblioteca Matplotlib

Es una biblioteca de Python para hacer gráficos 2D de matrices en Python. Aunque tiene su origen en emular los comandos de gráficos de MATLAB, es independiente de MATLAB y se puede utilizar en un objeto Pythonic, de forma orientada. Matplotlib está escrito principalmente en lenguaje Python pero, hace un uso considerable de NumPy y otro código de extensión para brindar el mejor rendimiento incluso para matrices grandes [41].

Matplotlib está diseñado con la filosofía de creación de gráficos simples con el uso de pocos comandos, o incluso solo uno [41].

1.3.14.4. Biblioteca Scikit-learn

Es una biblioteca de Python que contiene una variedad de algoritmos de aprendizaje automático. Esta librería es simple y eficiente en minería de datos y análisis de datos. Usa librerías NumPy, Scipy y Matplotlib [42].

1.3.14.5. Biblioteca Keras

Es una biblioteca de Python que permite implantar redes neuronales de alto nivel que son ejecutadas sobre TensorFlow. Es de fácil uso, y cada uno de sus módulos son independientes por lo cual se logra combinar y crear varios modelos de redes neuronales [43].

1.3.14.6. Biblioteca Tensorflow

Es una biblioteca de Python para el cálculo numérico distribuido utilizando gráficos de flujo de datos. Hace posible entrenar y ejecutar redes neuronales muy grandes de manera eficiente al distribuir los cálculos en potencialmente miles de servidores multi-GPU. TensorFlow se creó en Google y es compatible con muchas de sus aplicaciones de aprendizaje automático a gran escala. Es de código abierto desde noviembre de 2015 [42].

1.3.14.7. Biblioteca Pyqt5

Es una biblioteca de Python con uno de los mejores kits de herramientas de interfaz disponibles actualmente; es estable, maduro y completamente nativo. Controla todos los aspectos de los elementos de la interfaz de usuario [44].

2. METODOLOGÍA

En este capítulo se presenta la metodología utilizada para la generación de la base de datos histórica de eventos de fallas, la cual es usada para la construcción, entrenamiento y validación de una red neuronal artificial por medio del lenguaje de programación Python (ambiente SPYDER 3.6) para la predicción de fallas en líneas de transmisión.

2.1. SELECCIÓN Y ANÁLISIS DE DATOS DEL SISTEMA 39 BARRAS IEEE

El sistema IEEE de 39 barras se encuentra dentro del programa Power Factory DigSilent. Este programa computacional es una herramienta usada para el diseño y estudios de sistemas eléctricos de potencia.

El sistema IEEE de 39 barras también conocido como New-England Power System de 10 máquinas, consta de 34 líneas de transmisión, cuyos datos son usados para desarrollar el modelo de predicción.

Mediante el lenguaje de programación DPL se adquieren los datos de distancia, nivel de voltaje y nombre de las líneas de transmisión. Estos datos son exportados a un archivo de texto el cual se usará en la sección 2.2.

2.2. NÚMEROS ALEATORIOS Y PSEUDOALETORIOS

Los números aleatorios son esenciales en todas las técnicas de simulación. Un número aleatorio (uniforme) es una variable que tiene valores distribuidos uniformemente distribuidos ente intervalo de [0,1], es decir que la variable aleatoria puede tomar cualquier valor entre 0 y 1 con la misma probabilidad [12].

Los números aleatorios que son creados por una computadora utilizando algoritmos deterministas son llamados generadores de números aleatorios. Puesto que los números generados se ajusta a reglas matemáticas, no son considerados números aleatorios verdaderos y, en cambio, se denominan números pseudoaleatorios. Los números aleatorios de un generador deben poseer las siguientes características [12]:

- Aleatoriedad y distribución uniforme.
- Un gran período antes de que se repita la secuencia.
- Reproducibilidad para que se pueda repetir la misma secuencia.
- Eficiencia computacional en su creación.

Los algoritmos más populares para crear números aleatorios son los generadores congruentes.

El lenguaje de programación Python facilita la generación de números pseudoaleatorios mediante el uso de la librería Numpy la cual cumple con las características descritas anteriormente. Además, permite la generación de un solo valor o un vector de valores aleatorios.

2.2.1. CONVERSIÓN DE NÚMEROS ALEATORIOS UNIFORMES

La creación de una secuencia de números aleatorios usando la librería Numpy de Python y los datos de las líneas de transmisión son la base de datos de entrada necesarios para la simulación de Montecarlo.

A veces, los números aleatorios uniformes se pueden utilizar directamente para algunos tipos de problemas de simulación. En otros casos, deben convertirse en otras distribuciones no uniformes antes de que pueda comenzar el proceso de simulación. Los principales procedimientos utilizados para la conversión son [12]:

- Método de transformación inversa.
- Método de composición.
- Método de rechazo de aceptación.

El método de transformación inversa es el más eficiente, pero solo se puede usar si la distribución se puede invertir analíticamente [12].

Si la variable aleatoria tiene una función de densidad:

$$f(t) = \lambda e^{-\lambda t} \tag{2.1}$$

Donde:

λ: Tasa de falla $\lambda > 0$

t : Tiempo para la falla $t \geq 0$

Por el método de transformación inversa:

$$U = F(T) = 1 - e^{-\lambda t} \tag{2.2}$$

Entonces:

$$T = -\frac{1}{\lambda} \ln (1 - U) \tag{2.3}$$

Donde:

$F(T)$: Función de distribución de probabilidad acumulada.

U: Variable aleatoria distribuida uniformemente en el intervalo [0,1]

2.3. ESTIMACIÓN DE TASAS DE FALLAS PARA LAS LÍNEAS DE TRANSMISIÓN

Como se mencionó en la sección 1.3.3 existen ciertos problemas al momento de seleccionar una tasa de falla para los componentes de un SEP. Para resolver esos problemas, es común agrupar los datos de componentes similares. Aunque agrupar componentes se presenta como una solución factible, en ocasiones esto resulta en otra problemática donde se presentan situaciones como [14]:

- En el periodo de tiempo de los registros no existen fallas.
- Solo se cuenta con un dato, por lo cual no pueden usarse los métodos clásicos de estimación de parámetros o ajuste a una distribución de probabilidad.

El asumir que un componente es 100% confiable no es realista.

Al no contar con una base de datos de fallas no se puede encontrar la tasa de falla de las líneas de transmisión de sistema IEEE de 39 barras, por lo cual se usa el método de Poisson.

Se inicia con la generación de 34 números aleatorios entre 0 y 5, los cuales representan el número de fallas en las líneas de transmisión en un período de tiempo determinado.

Dependiendo del número de falla se selecciona el valor de la tasa de falla en base a la Tabla 1.2 para la línea de transmisión seleccionada.

2.4. BASE DE DATOS HISTÓRICA DE EVENTOS DE FALLA

Como se mencionó en la sección 2.2 los números aleatorios son esenciales en todas las técnicas de simulación. Para la creación de la base de datos se hace uso de la librería Numpy.

Para cumplir con las condiciones de la sección 1.3.3.1 y estimar la tasa de falla de las líneas de transmisión, se selecciona una tasa de reparación de 7.5 horas por reparación siendo despreciable en comparación de los tiempos de falla que se generaran.

Numpy de Python crea números aleatorios en manera vectorizada disminuyendo el uso de recursos del computador.

La base de datos histórica de eventos de falla contiene los datos de la línea de transmisión en falla, tipo de falla, distancia de falla, tiempo de falla y tiempo de reparación, finalmente se exporta en archivos que se utilizarán como datos de entrada a la RNA además de la creación de graficas.

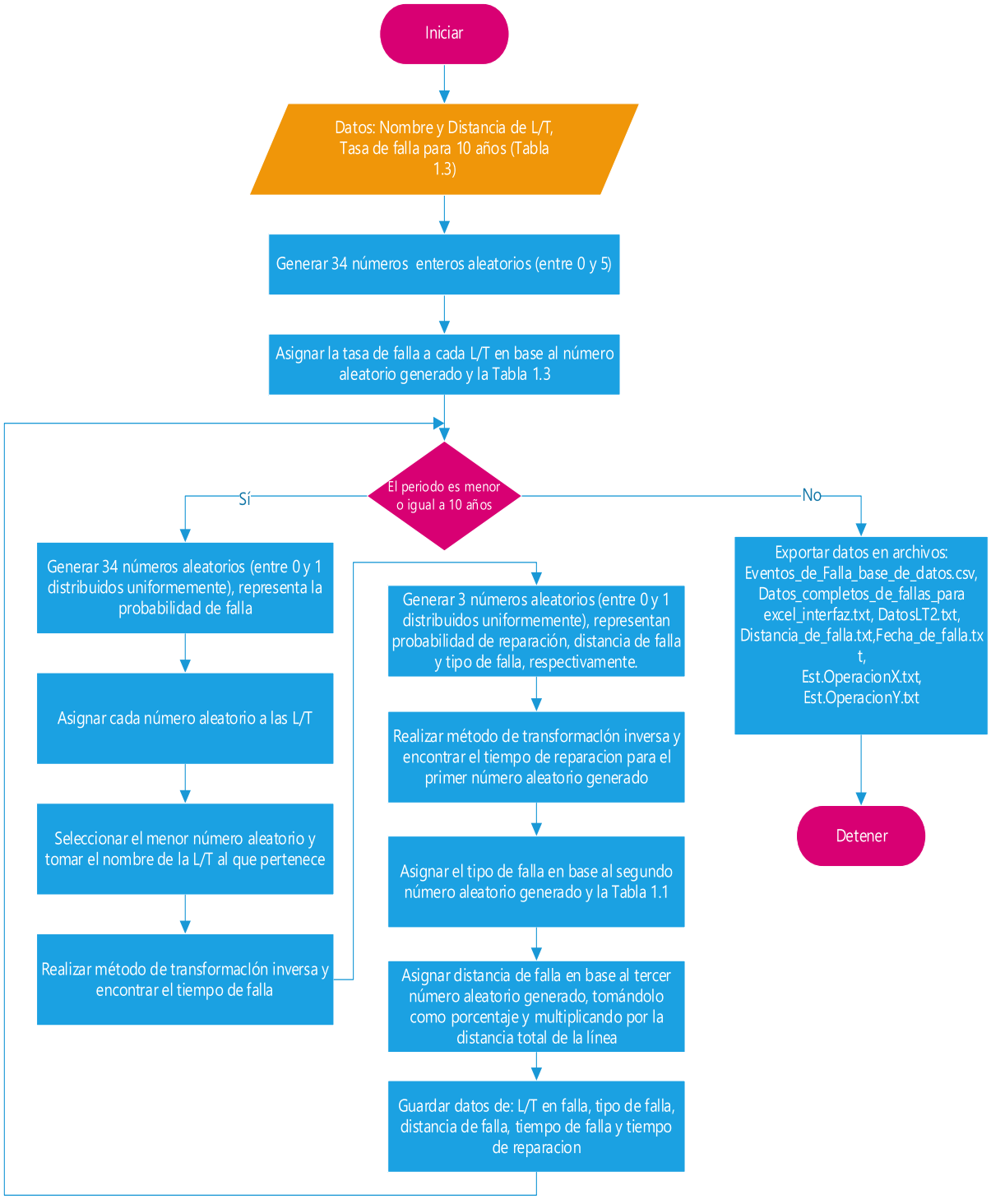


Figura 2.1. Diagrama de flujo para la creación de la base histórica de eventos de falla

2.5. DEFINICIÓN DE LA RED NEURONAL ARTIFICIAL PARA LA PREDICCIÓN DE FALLAS

En la sección 1.3.7.1 y 1.3.7.2 se indica la clasificación de las redes neuronales artificiales por su topología y su método de aprendizaje respectivamente.

La selección de una RNA se realiza en función de las características del problema a resolver. En base a los resultados de predicciones a lo largo de toda la literatura consultada, los libros [35], [42] y el curso de Sequences, Time Series and Prediction de [37] donde se presentan técnicas de como diseñar modelos de redes neuronales artificiales para la predicción de series, se hace uso de una red neuronal multicapa con aprendizaje supervisado.

Los distintos parámetros se los seleccionará mediante prueba y error hasta encontrar una RNA que de los mejores resultados y rendimiento.

2.5.1. NÚMERO DE CAPAS

La RNA puede estar compuesta de una capa de entrada, una capa de salida y una o varias capaz ocultas. Dependiendo de la función que va a realizar la red neuronal se puede ir incrementando las capas ocultas con el fin de lograr un modelo con un buen rendimiento. Las librerías TensorFlow y Keras permiten añadir diferentes tipos de red neuronal, como una red convolucional o una red neuronal recurrente.

En la mayoría de los casos con un número reducido de capas ocultas se logra obtener buenos resultados, pero al incrementarse la cantidad de datos a analizar y que no son lineales, se necesita incrementar este número de capas, por lo cual se examinarán diferentes configuraciones para cada RNA variando entre 1 y 2 capas para cada tipo de dato, distancia de falla, fecha de falla, línea en falla y tipo de falla.

2.5.2. NÚMERO DE NEURONAS POR CAPA

La capa de salida está definida por la cantidad de variables de salidas, al tener la distancia de falla como salida se define el número de neuronas en la capa de salida en 1.

La asignación de número de neuronas por capa no con lleva una regla, por lo cual se ir probando diferente número de neuronas por capa en la RNA y se escogerá la configuración que presente mejor rendimiento del modelo.

2.5.3. FUNCIÓN DE ACTIVACIÓN

A modo de encontrar la mejor predicción que se ajuste a los datos de entrada y tomando en cuenta que los datos están normalizados, se usarán funciones de activación ReLU y tangente hiperbólica.

2.6. ENTRENAMIENTO DE LA RED NEURONAL ARTIFICIAL

El mecanismo de aprendizaje que se seleccionó para esta red neuronal fue de aprendizaje supervisado. El método de propagación hacia atrás o backpropagation es empleado para el entrenamiento de la RNA. El criterio de validación para esta RNA fue el de error cuadrático medio (MSE). Se seleccionó este criterio de validación ya que, no es sensible a la presencia de anomalías o valores atípicos en los datos.

2.7. CASOS DE ESTUDIO DE LA RNA Y PROCEDIMIENTO

Se crearon diferentes casos de estudio con diferentes configuraciones de RNA a ser analizadas y comparadas para la predicción de fallas en las L/T. La comparación se la realiza con el error cuadrático medio (MSE) seleccionado la configuración que proporcione el mínimo valor del MSE.

Las variables de entrada son: distancia de falla, fecha de falla, línea en falla y tipo de falla. Cada una de estas variables entrará a un RNA independiente.

Tabla 2.1. Casos de estudio con diferentes números de capas ocultas

Configuración de estudio	Tipo de dato	Tipos de RNA a usarse en las capas ocultas	Número de capas ocultas
1	Distancia de falla	Perceptrón multicapa	1
2	Distancia de falla	Perceptrón multicapa	2
3	Fecha de falla	Perceptrón multicapa	1
4	Fecha de falla	Perceptrón multicapa	2
5	Línea en falla	Perceptrón multicapa	1
6	Línea en falla	Perceptrón multicapa	2
7	Tipo de falla	Perceptrón multicapa	1
8	Tipo de falla	Perceptrón multicapa	2

A manera de mostrar la importancia de la normalización de la información, se usarán los datos de distancia de falla sin normalizar.

Para realizar la predicción de fallas primero se debe ejecutar el archivo BASE_DE_DATOS_EVENT_FALLAS.py. Segundo se debe copiar los archivos generados en la carpeta que aloje a la interfaz gráfica en este caso ANEXO A.

Tabla 2.2. Casos de estudio para datos no normalizados de distancia de falla

Configuración de estudio	Tipo de dato	Tipos de RNA a usarse en las capas ocultas.	Número de capas ocultas
9	Distancia de falla	Perceptrón multicapa	1
10	Distancia de falla	Perceptrón multicapa	2
11	Distancia de falla	RNN + Perceptrón multicapa	3
12	Distancia de falla	RNN + Perceptrón multicapa	5

Los datos que ingresan a la correspondiente RNA ingresan de la siguiente manera:

- **Distancia de falla:** La distancia de falla generada en la base de datos de eventos de falla ingresa en kilómetros a la RNA.
- **Fecha de falla:** La fecha de falla generada en la base de datos de eventos de falla es transformada a horas y es ingresada a la RNA al finalizar se la transforma a una fecha.
- **Línea en falla:** El nombre de la Línea en falla generada en la base de datos de eventos de falla es transformada a un número, este número es el digito de su nombre. Ejemplo: Línea 3, ingresa a la RNA como el número 3.
- **Tipo de falla:** El tipo de falla generada en la base de datos de eventos de falla es transformada a un número mediante su designación. Falla monofásica:1, Falla bifásica:2, Falla bifásica a tierra: 3, Falla trifásica:4. Por lo cual a la RNA ingresa un número.

3. RESULTADOS Y DISCUSIÓN

En este capítulo se presentan los resultados obtenidos de la generación de la base de histórica de ventos de falla, los resultados de la RNA, el entrenamiento y validación de las diferentes configuraciones de la red neuronal artificial y su interfaz gráfica. Para el aprendizaje y validación se usan los datos de eventos de falla generados mediante la Simulación de Montecarlo y números pseudoaleatorios. Los datos de eventos de falla constan de: distancia de falla, fecha de falla, línea en falla y tipo de falla para un periodo de 10 años.

En primera instancia se muestran y compraran los resultados con los datos de eventos de falla no normalizados y normalizados. Una vez descritas las diferentes configuraciones con datos normalizados se selecciona la de mejor rendimiento, esto en base al error cuadrático medio o MSE. Seleccionada la configuración de la RNA se la incluye dentro de una interfaz gráfica para mejor visualización de los resultados. Cabe mencionar que, para cada tipo de dato, distancia de falla, fecha de falla, línea en falla y tipo de falla se selecciona una configuración diferente de red neuronal artificial.

3.1. SOFTWARE DEL MODELO DE PREDICCIÓN DE FALLAS

El código fuente de la programación se detalla en el ANEXO A del presente documento.

El software del modelo de predicción de fallas se inicia al ejecutar interfaz_final.py, consta de tres ventanas. La primera ventana es la ventana principal. Segunda ventana donde se puede visualizar la información de los eventos de falla generados mediante la Simulación de Montecarlo y números pseudoaleatorios, en la cual se puede visualizar de forma gráfica todas las fallas y el estado de operación de las L/T. Tercera ventana consta de las RNA para distancia de falla, fecha de falla, línea en falla y tipo de falla, en la cual también se puede visualizar de forma gráfica y numérica los resultados de la red neuronal artificial.

3.1.1. VENTANA INICIAL

Ejecutado el archivo MODELO_FINAL_DE_PREDICCION.py permite al usuario el acceso a la ventana inicial donde se muestra el nombre del proyecto de titulación, nombre del autor y con un botón (INGRESAR) para el acceso a la base de datos histórica de eventos de falla.

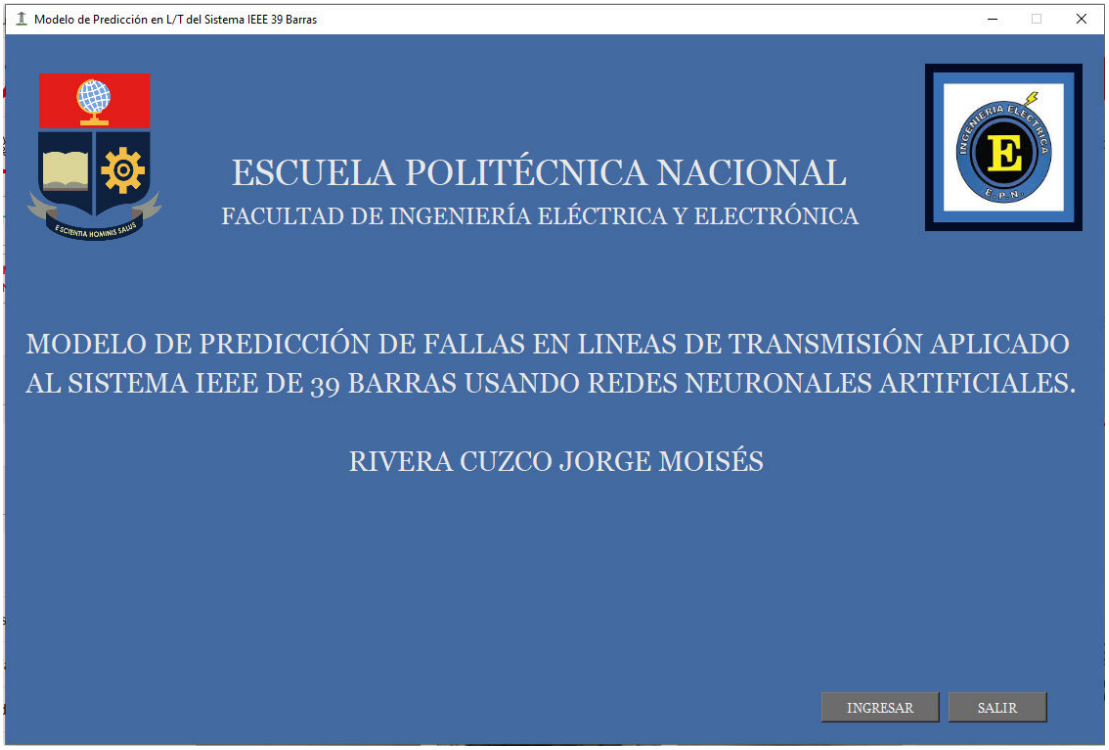


Figura 3.1. Ventana inicial de la interfaz gráfica

3.1.2. VENTANA DE INFORMACIÓN DE EVENTOS DE FALLA

Una vez seleccionado la opción de INGRESAR de la ventana inicial, se muestra la ventana donde se puede visualizar la información de los datos de eventos de falla (ver Figura 3.2).

Al presionar el botón Mostrar datos de fallas, se muestran las fechas de fallas, distancias de falla, L/T en falla y tipo de falla en la parte superior izquierda.

Al presionar el botón Mostrar datos de L/T, se muestran la información de las líneas de transmisión, N° de L/T, nombre de la L/T, distancia, tasa de falla y tasa de reparación.

Al presionar el botón Mostrar falla se muestra de manera gráfica la distancia de falla vs número de falla, cabe mencionar que al dar doble clic en la columna de Falla No (Número de falla) se señala la falla en la gráfica mencionada (ver Figura 3.3). Con el mismo botón se muestra los estados de operación de las líneas de transmisión donde 0 indica fuera de servicio y 1 indica operación de la L/T.

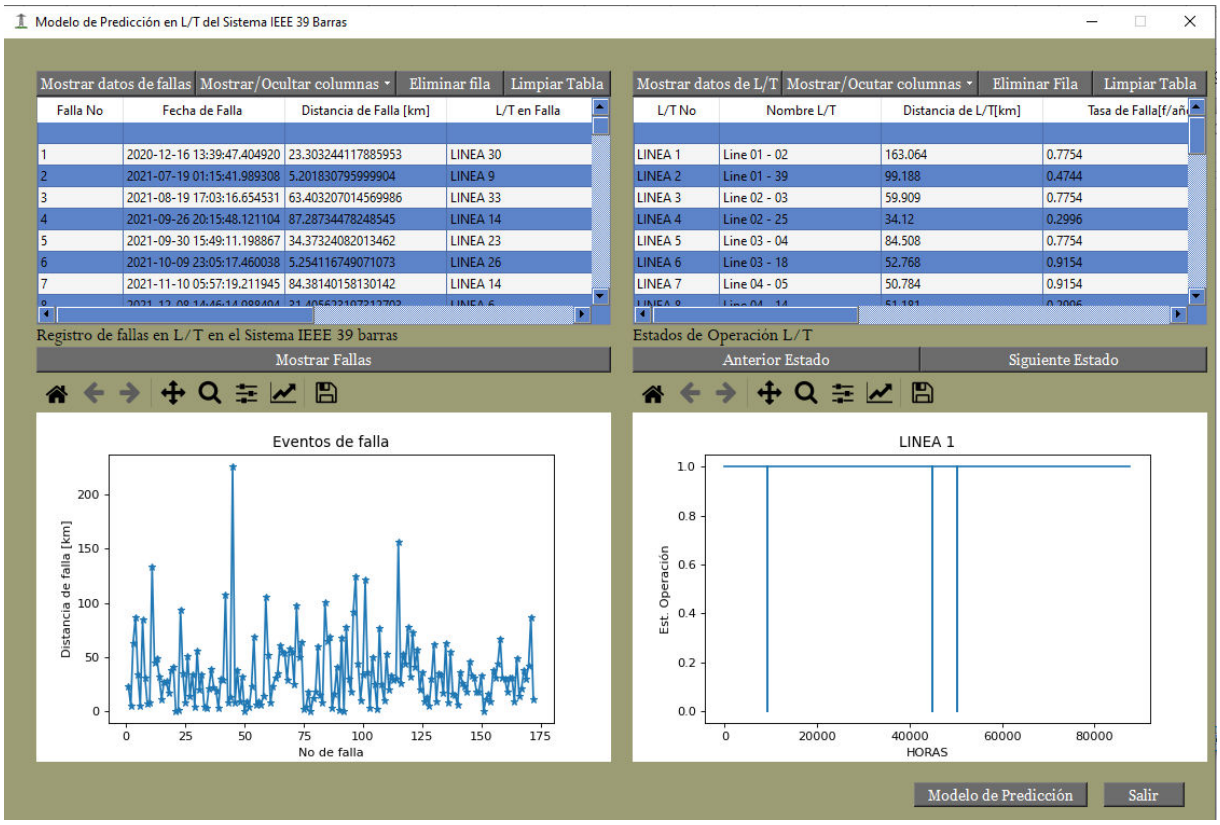


Figura 3.2. Ventana de información de L/T de la interfaz gráfica

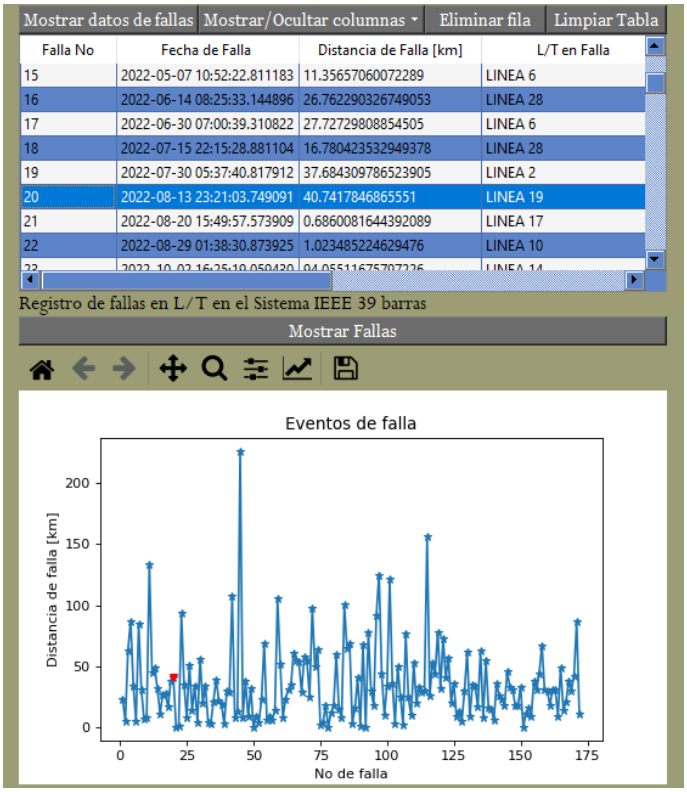


Figura 3.3. Ventana de información de la falla 20

3.1.3. VENTANA DE LA RED NEURONAL ARTIFICIAL

Una vez seleccionado la opción de Modelo de Predicción de la ventana de información, se muestra la ventana donde se puede iniciar con la red neuronal artificial, esto al presionar el botón Iniciar RNA (ver Figura 3.4).

Una vez presionado el botón Iniciar RNA, la red neuronal toma los datos del archivo Eventos_de_Falla_base_de_datos.csv, donde:

- La columna 2 predice la distancia de falla de las L/T.
- La columna 3 predice la L/T en falla
- La columna 5 predice tipo de falla.
- La columna 6 predice la fecha de falla.

Presionando los botones de Distancia, Fecha, Línea y Tipo Falla se van mostrando las gráficas de los datos reales y la predicción de cada uno (ver Figura 3.5, Figura 3.6, Figura 3.7, Figura 3. 8). De manera similar que la ventana de información al dar doble clic izquierdo en la falla de la columna Falla No se marca la falla seleccionada en la gráfica correspondiente.

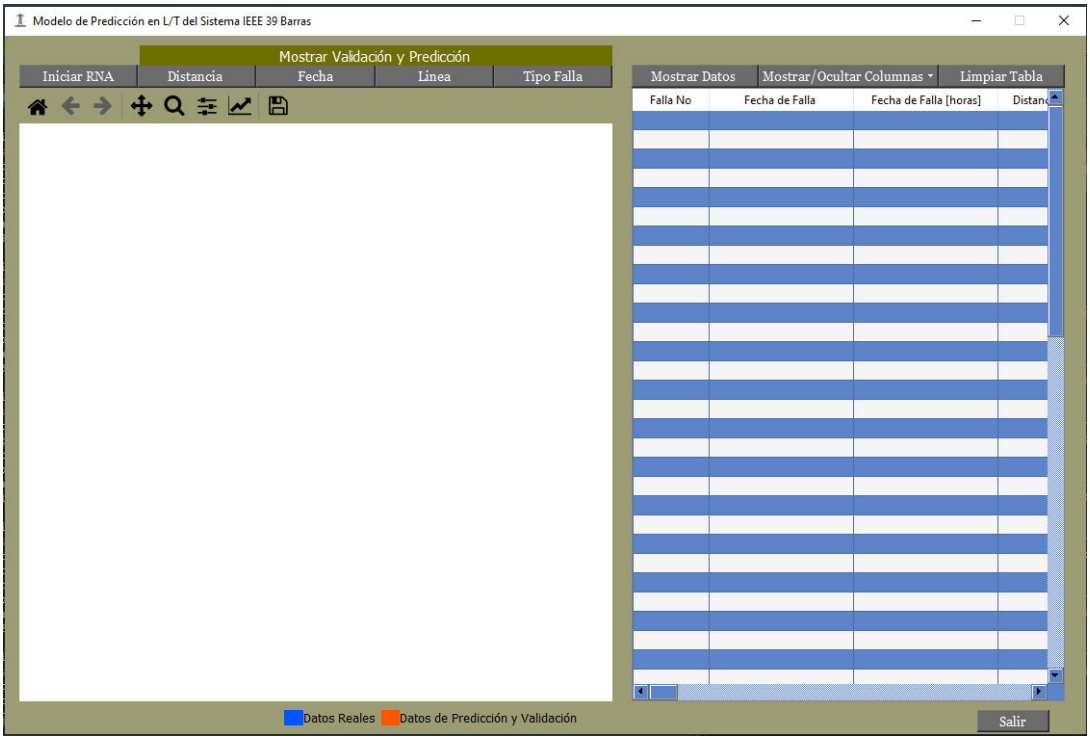


Figura 3.4. Ventana de RNA

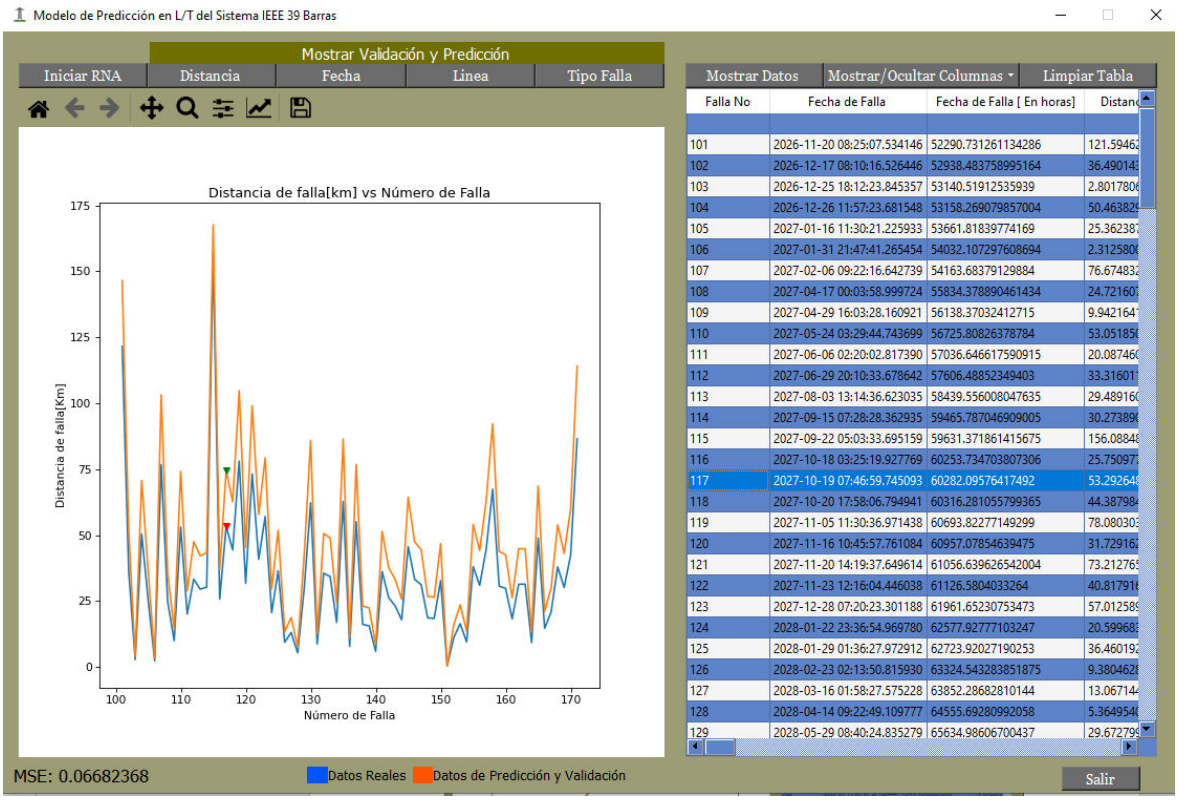


Figura 3.5. RNA para distancia de falla

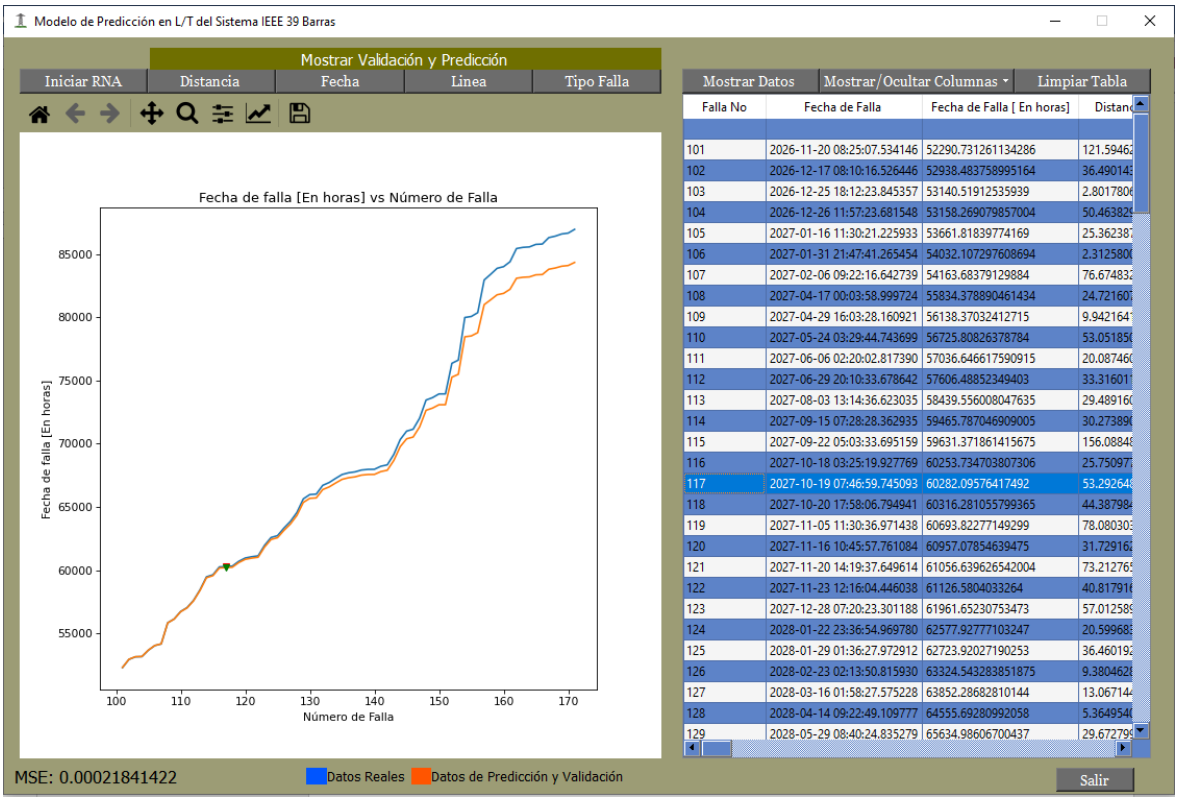


Figura 3.6. RNA para fecha de falla

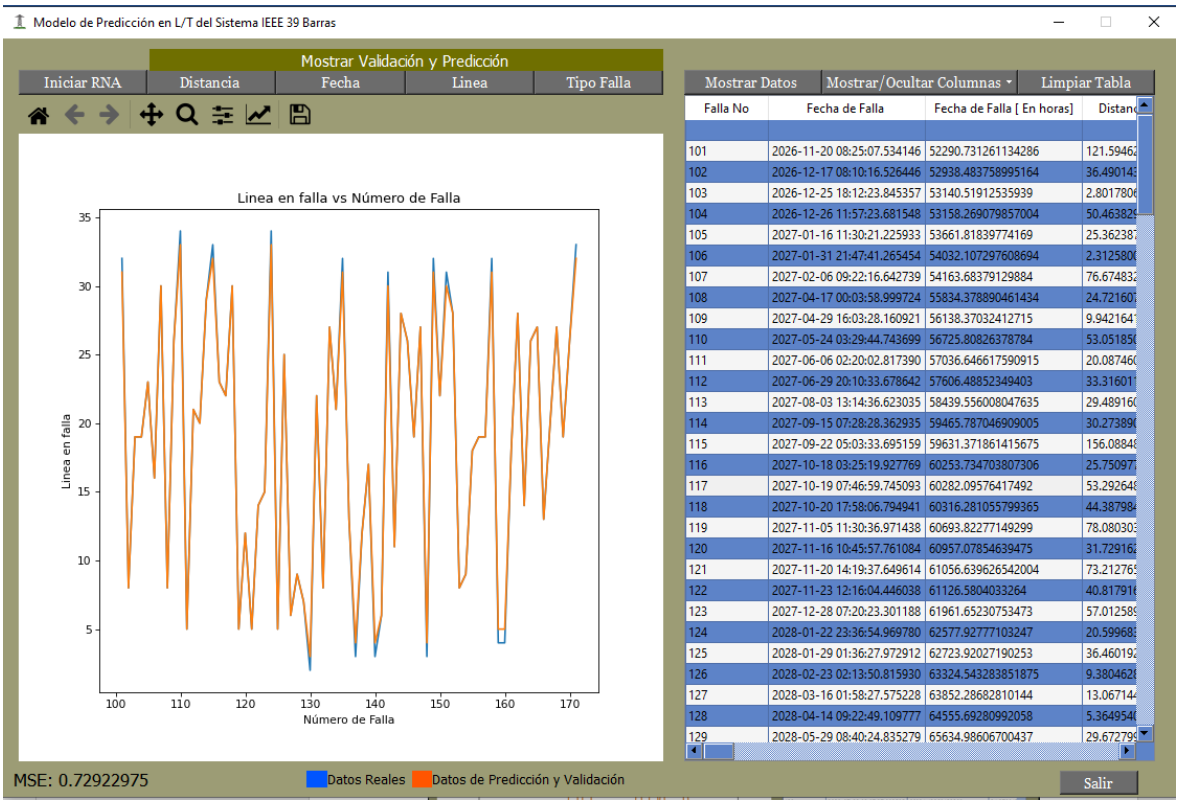


Figura 3.7. RNA para L/T en falla

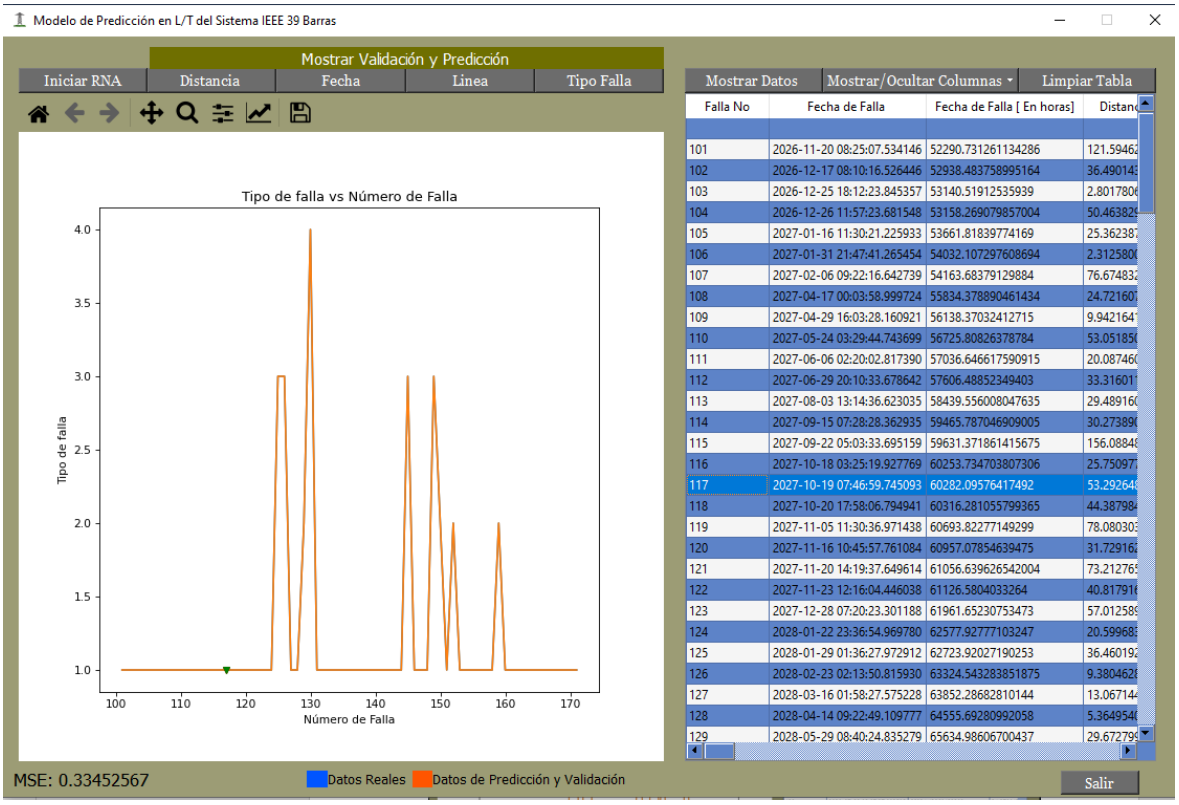


Figura 3. 8. RNA para tipo de falla

3.2. EVENTOS DE FALLA

El código fuente de la programación se detalla en el ANEXO B del presente documento.

La generación de la base de datos de eventos de falla se la realiza al ejecutar el archivo BASE_DE_DATOS_EVENT_FALLAS.py. Para el periodo seleccionado de 10 años se obtuvieron 172 fallas, este número de fallas depende de la tasa de falla y tasa de reparación de cada L/T.

Al finalizar el archivo BASE_DE_DATOS_EVENT_FALLAS.py se generan diferentes archivos, cuyo fin es ser usados por la red neuronal artificial como información de entrada a la misma y además para las correspondientes graficas.

Datos_completos_de_fallas_para Excel.xslx y Eventos_de_Falla_base_de_datos.csv: Muestra la fecha de falla, distancia de falla, línea en falla, tipo de falla, falla en horas como se muestra en la Figura 3.9.

DatosLT2.txt: Se alojan los datos de nombres, distancias, tasa de falla y tasa de reparación de las L/T.

Est.OperacionX.txt y Est.OperacionY.txt: Se alojan los tiempos y estado de operación respectivamente de cada L/T.

Cabe mencionar que el archivo BASE_DE_DATOS_EVENT_FALLAS.py tomara como fecha inicial la fecha y hora en la que sea ejecutad. Sin embargo, esto no influye en el número de fallas generada ya que estas dependen de la tasa de falla, tasa de reparación y periodo como se mencionó anteriormente.

La Figura 3.10 y Figura 3.11 muestran los estados de operación de las líneas de transmisión 3 y 7 respectivamente las cuales se muestran al finalizar la ejecución del archivo BASE_DE_DATOS_EVENT_FALLAS.py.

	Fecha de falla	Distancia de falla	Linea en falla	Tipo de falla	Denominacion de Tip. Falla	Horas en operación hasta la falla	Fecha inicial
0	2020-12-16 19:02:55	23,30324412	LINEA 30	Falla Bifasica	2	335,9756697	2020-12-02 19:04:22
1	2021-07-19 06:38:49	5,201830796	LINEA 9	Falla Monofasica	1	5483,574165	2020-12-02 19:04:22
2	2021-08-19 22:26:24	63,40320701	LINEA 33	Falla Monofasica	1	6243,367128	2020-12-02 19:04:22
3	2021-09-27 01:38:55	87,28734478	LINEA 14	Falla Monofasica	1	7158,575869	2020-12-02 19:04:22
4	2021-09-30 21:12:18	34,37324082	LINEA 23	Falla Monofasica	1	7250,132279	2020-12-02 19:04:22
5	2021-10-10 04:28:25	5,254116749	LINEA 26	Falla Monofasica	1	7473,400685	2020-12-02 19:04:22
6	2021-11-10 11:20:26	84,38140158	LINEA 14	Falla Monofasica	1	8224,267838	2020-12-02 19:04:22
7	2021-12-08 20:09:22	31,4056232	LINEA 6	Falla Bifasica	2	8905,083332	2020-12-02 19:04:22
8	2021-12-15 14:48:09	7,045887966	LINEA 21	Falla Monofasica	1	9067,729707	2020-12-02 19:04:22

Figura 3.9. Primeras 9 fallas de la base de datos

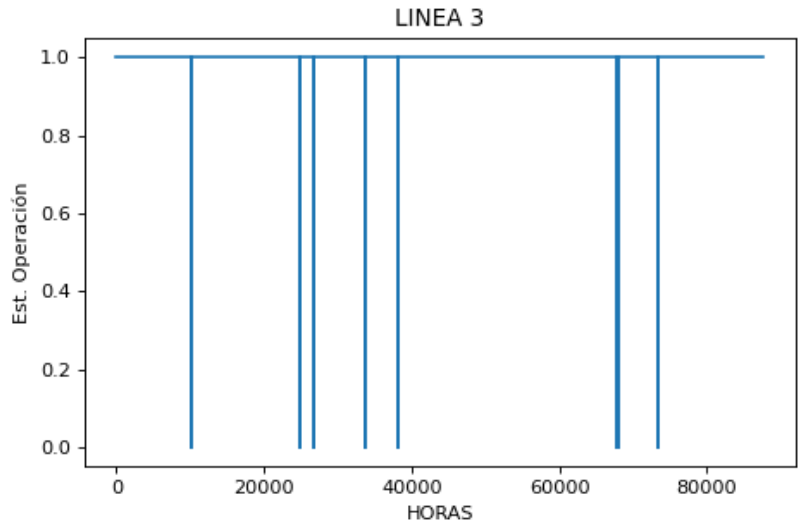


Figura 3.10. Estados de operación de la Línea 3

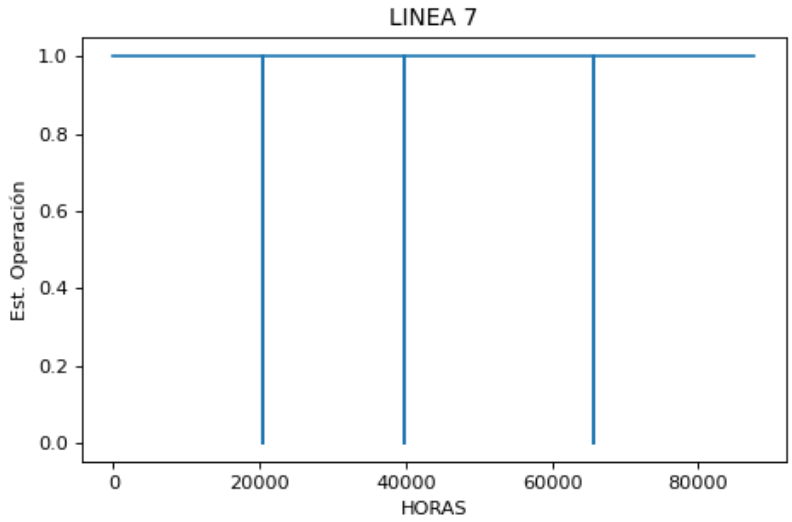


Figura 3.11. Estados de operación de la línea 7

3.3. RED NEURONAL ARTIFICIAL Y DATOS NO NORMALIZADOS

La normalización de los datos de entrada en el preprocesamiento de datos influye en el desempeño de la RNA, al tener altos valores en los datos se pierde precisión en la predicción, por lo cual se debe incrementar el número de capas y número de neuronas en la RNA.

Como ejemplo se tiene la Figura 3.12 y la Tabla 3.1, donde la predicción de la distancia de falla es mala ya que se muestra una línea recta. Su MSE tiene un valor de 1556.8188 lo cual es muy elevado, para mejorar este resultado se puede incrementar en número de capas ocultas y numero de neuronas.

Tabla 3.1. Configuración de estudio 9

Tipo de dato: Distancia de falla	**Tipo de RNA**	**Número de neuronas**	**Función de activación**
Capa de entrada	Perceptrón multicapa	1	-----
Capa oculta 1		5	relu
Capa de salida		1	------

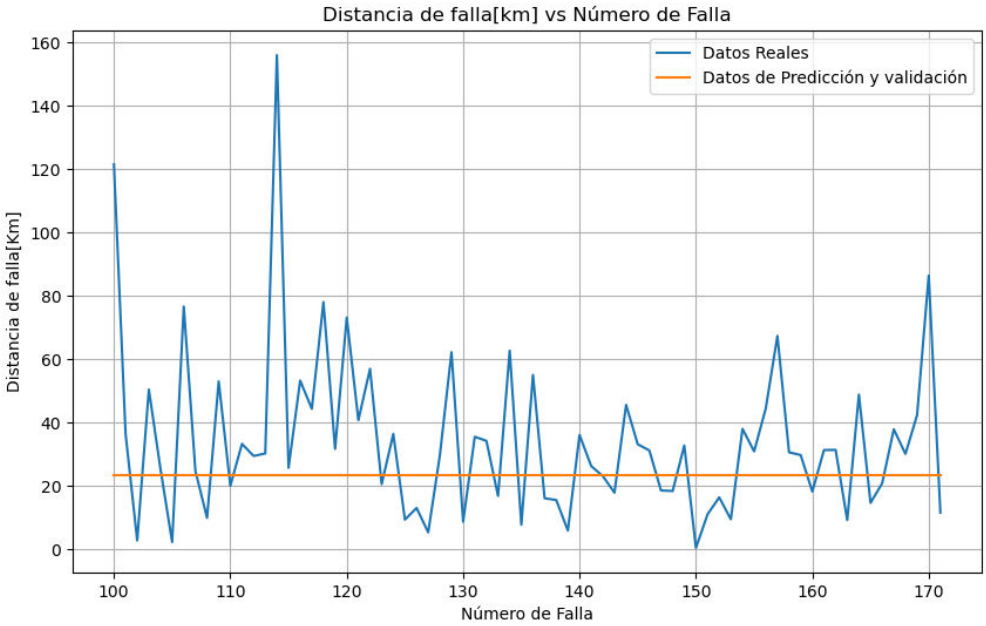


Figura 3.12. Gráfica de la configuración de estudio 9

Al incrementar una capa oculta y el número de neuronas como se indica en la Tabla 3.2 se puede apreciar que la predicción de los datos cambia de una línea recta y disminuye su MSE a 1465.2505 el cual es menor que la configuración 9. Sin embargo, la predicción no se ajusta a los valores reales.

Tabla 3.2. Configuración de estudio 10

Tipo de dato: Distancia de falla	**Tipo de RNA**	**Número de neuronas**	**Función de activación**
Capa de entrada	Perceptrón multicapa	1	-----
Capa oculta 1		5	relu
Capa oculta 2		10	tanh
Capa de salida		1	-----

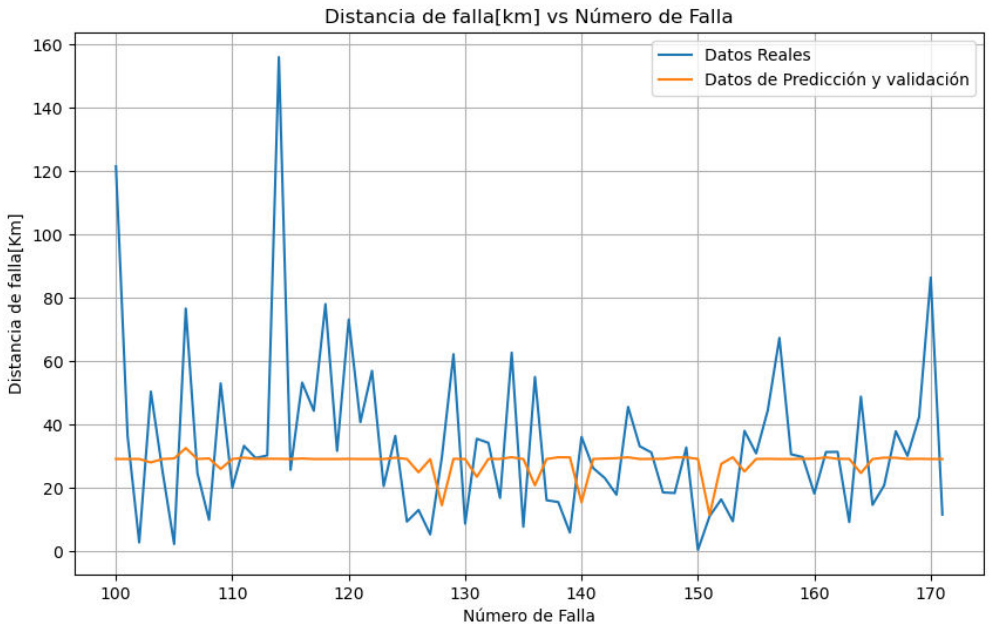


Figura 3.13. Gráfica de la configuración de estudio 10

Continuando con el incremento de las capas ocultas y número de neuronas se tiene la configuración de estudio 11. La Tabla 3.3 detalla la configuración de RNA usada la cual usa una red neuronal recurrente en la capa oculta 1 y perceptrón multicapa en para las siguientes capas ocultas. Como se muestra en la Figura 3.13 la predicción de datos trata de ajustarse a los datos reales teniendo un MSE de 1438.0175 es cual es menor que la configuración de estudio 10.

Tabla 3.3. Configuración de estudio 11

Tipo de dato: Distancia de falla	**Tipo de RNA**	**Número de neuronas**	**Función de activación**
Capa de entrada	RNN + Perceptrón multicapa	1	-----
Capa oculta 1		5	relu
Capa oculta 2		10	relu
Capa oculta 3		15	tanh
Capa de salida		1	-----

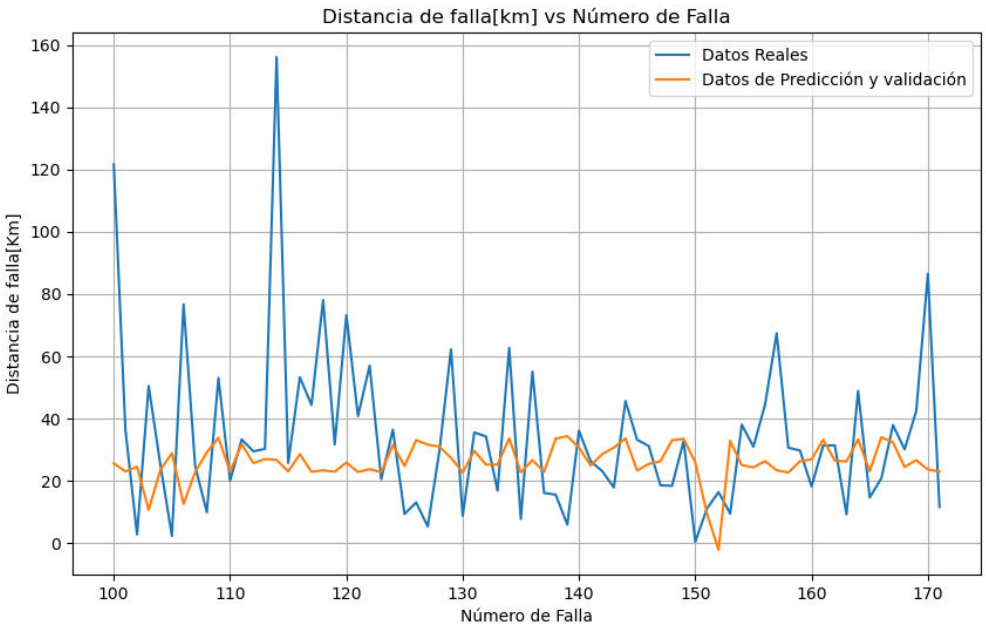


Figura 3.14. Gráfica de la configuración de estudio 11

La configuración 12 se detalla en la Tabla 3.4 en la cual las 3 primeras capas ocultas son redes neuronales recurrentes, de igual manera para esta configuración se incrementó las capas ocultas y el número de neuronas, obteniendo un MSE de 1458.7712, el cual tiene un incremento en comparación con la configuración de estudio 11, pero los datos tratan de mejor manera de ajustarse a los datos reales como se muestra en la Figura 3.14.

Tabla 3.4. Configuración de estudio 12

Tipo de dato: Distancia de falla	**Tipo de RNA**	**Número de neuronas**	**Función de activación**
Capa de entrada	RNN + Perceptrón multicapa	1	-----
Capa oculta 1		5	relu
Capa oculta 2		10	relu
Capa oculta 3		15	tanh
Capa oculta 4		20	tanh
Capa oculta 5		25	tanh
Capa de salida		1	-----

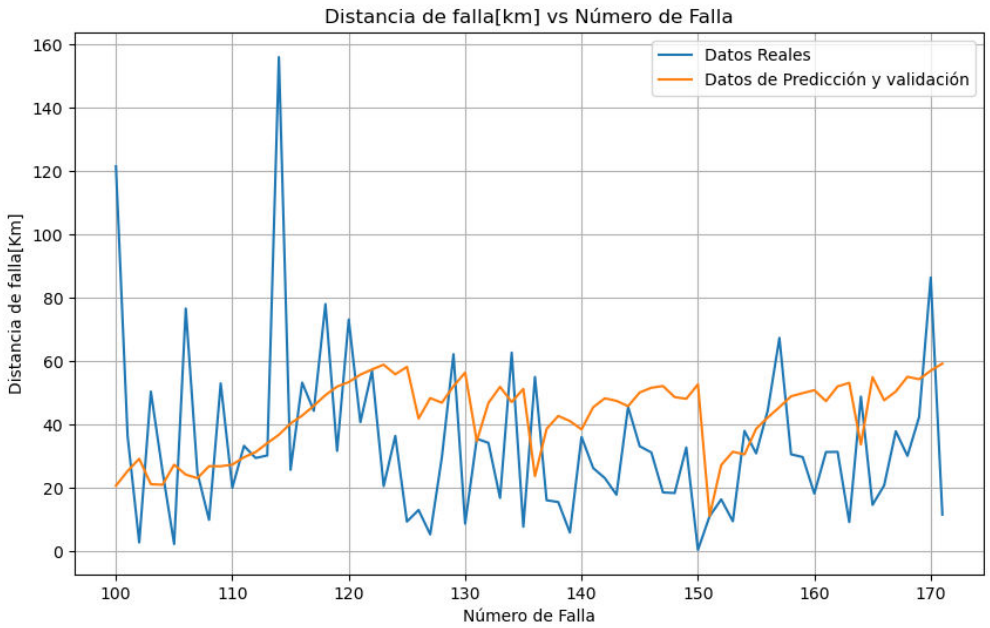


Figura 3.15. Gráfica de la configuración de estudio 12

Como se puede apreciar en los resultados, al usar datos no normalizados como información de entrada a una red neuronal artificial su MSE es muy elevado puesto que este indicador amplifica los errores mayores debido a que calcula el cuadrado del error. Además se observa en la Figura 3.12, Figura 3.13, Figura 3.14 y Figura 3.15 el pronóstico de los datos no se ajusta de manera adecuada a los datos reales.

El número de capas ocultas y neuronas en cada capa se puede ir incrementando para obtener mejores resultados, pero esto influirá en el tiempo de procesamiento del computador.

3.4. RED NEURONAL ARTIFICIAL Y DATOS NORMALIZADOS

Como resultado del uso de datos normalizados se evidencia que la predicción de la RNA se ajusta mejor a los datos reales ingresados a la RNA. Para cada configuración de estudio se constata que su respectivo MSE es mucho menor a las configuraciones de estudio de la sección 3.3.

Para la configuración de estudio 1 se obtuvo un MSE 2.6669, que en comparación con la configuración de estudio 12 su MSE es de 1458.7712. La predicción de datos para este caso de estudio 1 no se ajusta a los datos reales para lo cual se procede a incrementar una capa y el número de neuronas como se muestra en la Tabla 3.5.

Tabla 3.5. Configuración de estudio 1

Tipo de dato: Distancia de falla	**Tipo de RNA**	**Número de neuronas**	**Función de activación**
Capa de entrada	Perceptrón multicapa	1	-----
Capa oculta 1		5	relu
Capa de salida		1	------

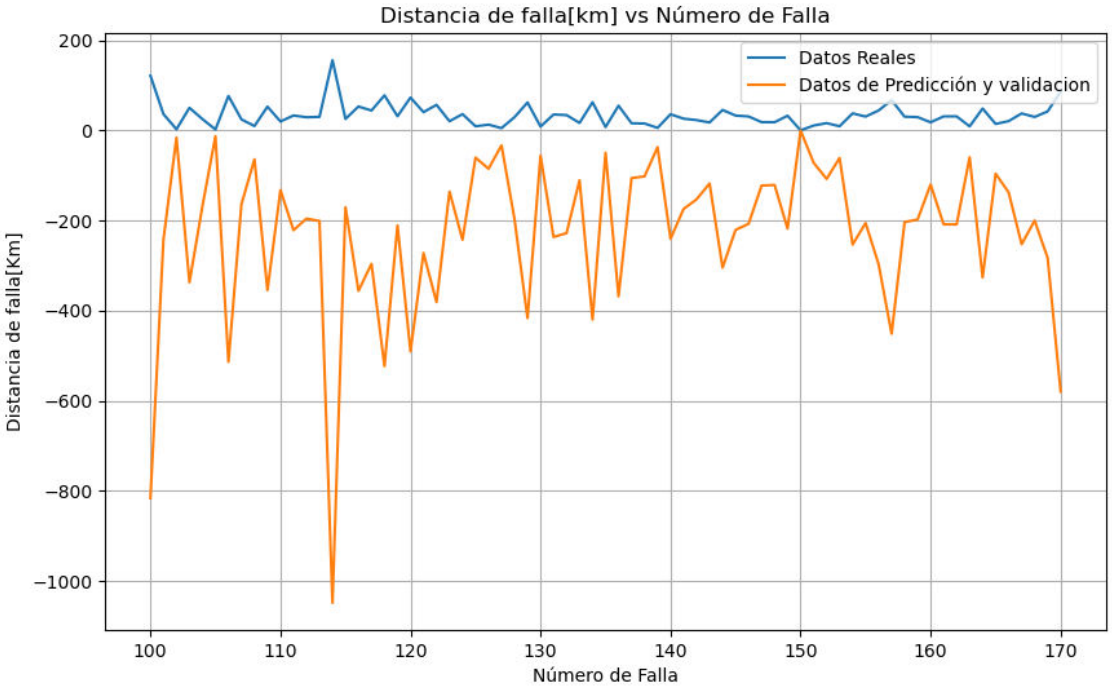


Figura 3.16. Gráfica de la configuración de estudio 1

Con un MSE de 0.0668 la configuración de estudio 2 presenta el menor valor en comparación con las configuraciones anteriores para el dato de distancia de falla. Como se puede apreciar en la Figura 3.17 la predicción se ajusta de manera adecuada a los datos reales, la configuración para esta RNA se detalla en la Tabla 3.6.

Tabla 3.6. Configuración de estudio 2

Tipo de dato: Distancia de falla	**Tipo de RNA**	**Número de neuronas**	**Función de activación**
Capa de entrada	Perceptrón multicapa	1	-----
Capa oculta 1		2	relu
Capa oculta 2		5	tanh
Capa de salida		1	-----

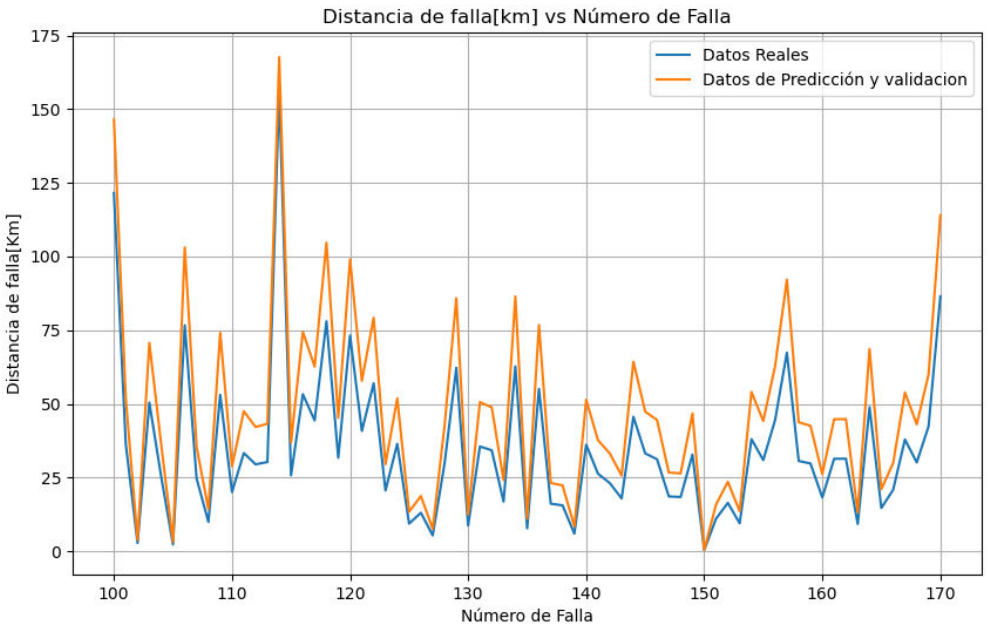


Figura 3.17. Gráfica de la configuración de estudio 2

Como se puede ver en la Figura 3.18 la predicción de la fecha de falla es mala ya que toma valores negativos lo cual es incorrecto, esto puede deberse a su función de activación o el rango de normalización, sin embargo, el MSE es de 5.2947 que es un valor bajo de error. Se procede a incrementar el número de capas y neuronas para la configuración de estudio 4.

Tabla 3.7. Configuración de estudio 3

Tipo de dato: Fecha de falla	**Tipo de RNA**	**Número de neuronas**	**Función de activación**
Capa de entrada	Perceptrón multicapa	1	-----
Capa oculta 1		5	tanh
Capa de salida		1	------

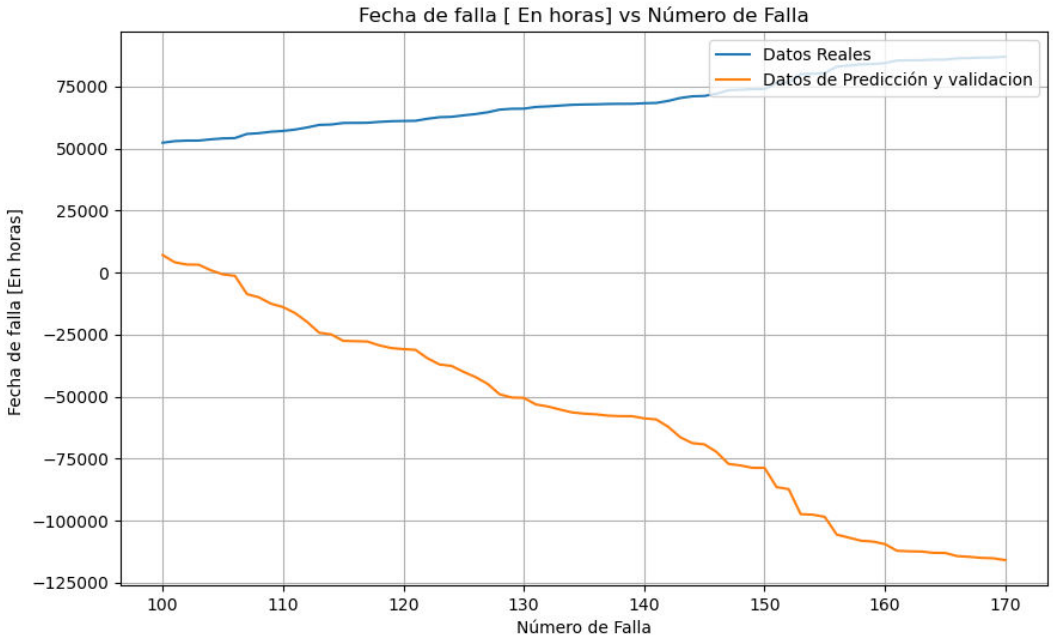


Figura 3.18. Gráfica de la configuración de estudio 3

De la Figura 3.19, se aprecia que la predicción de la fecha de falla es buena ya que no toma valores negativos como en la configuración de estudio 3. Con un MSE de 2.1841e-04 indica que los cambios realizados son correctos, incrementar una capa oculta y colocar mayor número de neuronas y mezclar las funciones de activación, la Tabla 3.8 muestra la configuración de estudio 4.

Tabla 3.8. Configuración de estudio 4

Tipo de dato: Fecha de falla	**Tipo de RNA**	**Número de neuronas**	**Función de activación**
Capa de entrada	Perceptrón multicapa	1	-----
Capa oculta 1		1	relu
Capa oculta 2		18	tanh
Capa de salida		1	-----

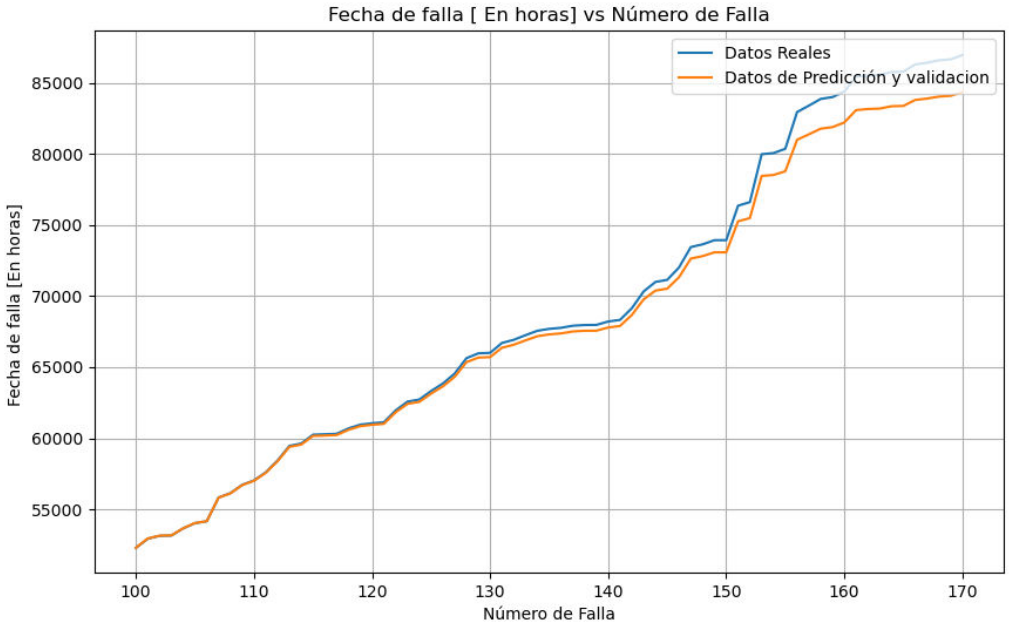


Figura 3.19. Gráfica de la configuración de estudio 4

En la Figura 3.20 se evidencia que la predicción de línea en falla es mala ya que toma valores negativos lo cual es incorrecto, esto puede deberse a su función de activación o el rango de normalización, sin embargo, su MSE es de 4.0361 el cual es un valor bajo de error. Se procede a incrementar el número de capas y neuronas para la configuración de estudio 6.

Tabla 3.9. Configuración de estudio 5

Tipo de dato: Línea en falla	**Tipo de RNA**	**Número de neuronas**	**Función de activación**
Capa de entrada	Perceptrón multicapa	1	-----
Capa oculta 1		5	relu
Capa de salida		1	------

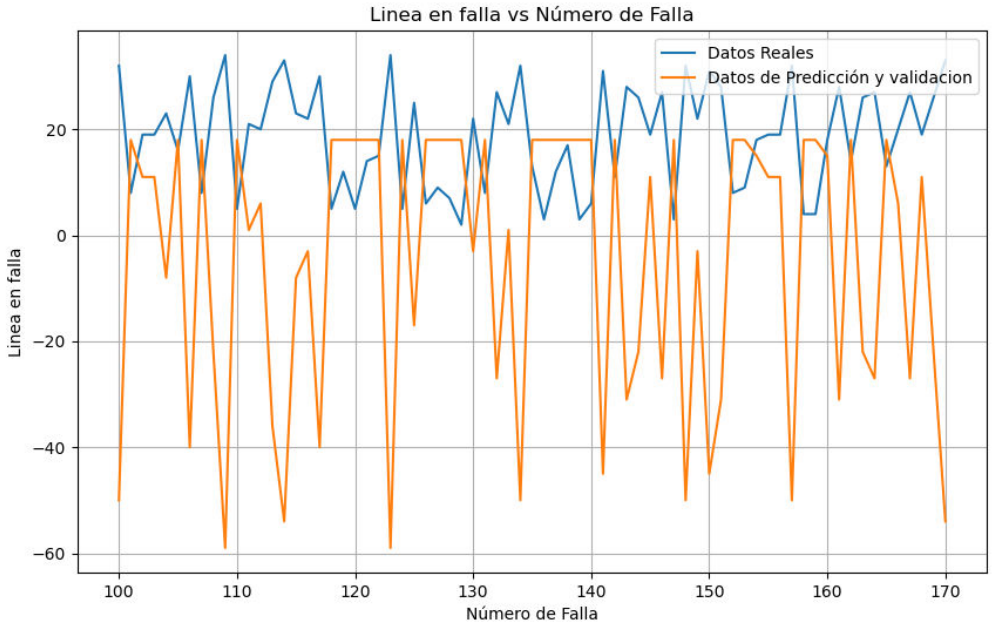


Figura 3.20. Gráfica de la configuración de estudio 5

La Figura 3.21 muestra que la predicción de línea en falla es buena ya que no toma valores negativos como en la configuración de estudio 3. Con un MSE de 0.7292 indica que los cambios realizados son correctos, incrementar una capa oculta y colocar mayor número de neuronas y mezclar las funciones de activación, la Tabla 3.10 muestra la configuración de estudio 6.

Tabla 3.10. Configuración de estudio 6

Tipo de dato: Línea en falla	**Tipo de RNA**	**Número de neuronas**	**Función de activación**
Capa de entrada	Perceptrón multicapa	1	-----
Capa oculta 1		1	tanh
Capa oculta 2		18	tanh
Capa de salida		1	-----

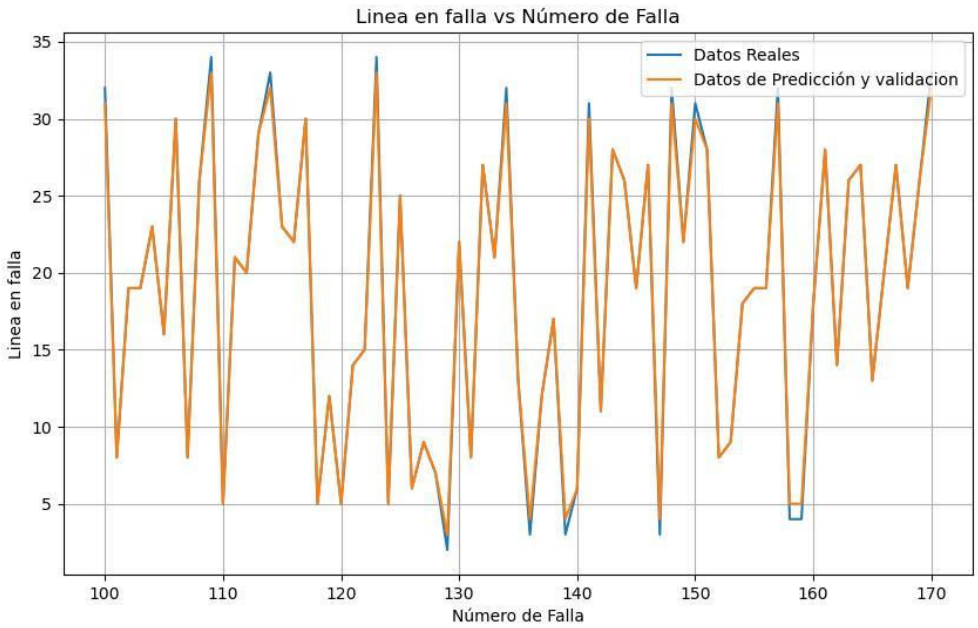


Figura 3.21. Gráfica de la configuración de estudio 6

En la Figura 3.22 se aprecia que la predicción de línea en falla es mala ya que toma valores negativos lo cual es incorrecto, además no se ajusta a los valores reales, esto puede deberse a su función de activación o el rango de normalización, sin embargo, su MSE es de 0.8014 el cual es un valor bajo de error. Se procede a incrementar el número de capas y neuronas para la configuración de estudio 8.

Tabla 3.11. Configuración de estudio 7

Tipo de dato: Tipo de falla	**Tipo de RNA**	**Número de neuronas**	**Función de activación**
Capa de entrada	Perceptrón multicapa	1	-----
Capa oculta 1		5	relu
Capa de salida		1	------

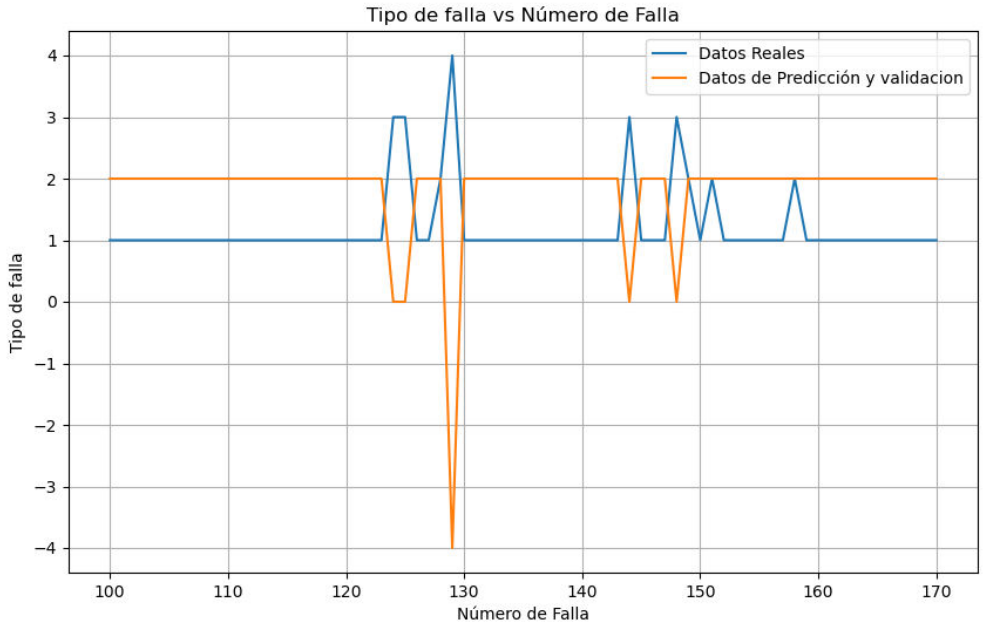


Figura 3.22. Gráfica de la configuración de estudio 7

En la Figura 3.23 se evidencia una buena predicción de línea en falla es buena ya que no toma valores negativos como en la configuración de estudio 3 además de ajustarse de mejor manera a los datos reales. Con un MSE de 0.3345 indica que los cambios realizados son correctos, incrementar una capa oculta y colocar mayor número de neuronas y mezclar las funciones de activación, la Tabla 3.12 muestra la configuración de estudio 8.

Tabla 3.12. Configuración de estudio 8

Tipo de dato: Tipo de falla	**Tipo de RNA**	**Número de neuronas**	**Función de activación**
Capa de entrada	Perceptrón multicapa	1	-----
Capa oculta 1		1	tanh
Capa oculta 2		18	tanh
Capa de salida		1	-----

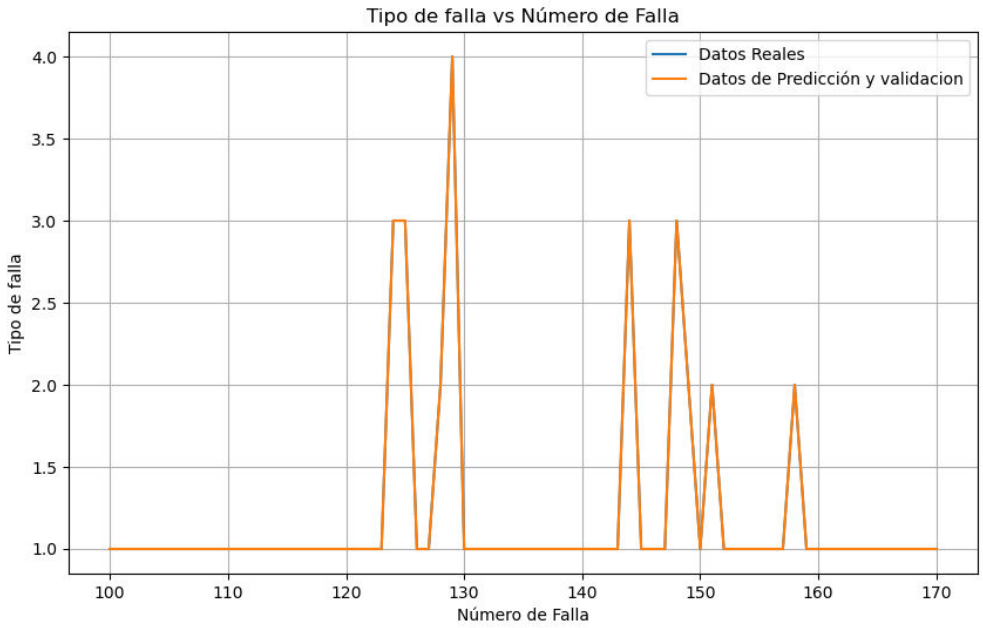


Figura 3.23. Gráfica de la configuración de estudio 8

3.5. TASA DE APRENDIZAJE DE LA RED NEURONAL

Posterior a la selección del número de capaz y número de neuronas obtiene la gráfica Pérdidas vs Tasa de aprendizaje, la cual permitió seleccionar la tasa de aprendizaje para cada red neuronal artificial. La Figura 3. 24 muestra como las pérdidas de información disminuyen al incrementar su tasa de aprendizaje en la configuración de estudio 2. Cabe mencionar que al variar el número de neuronas o número de capas su gráfica de aprendizaje cambia.

Para las Figura 3. ***25*** y Figura 3. ***26*** se modificó el número de neuronas. En la Figura 3. ***25*** se incrementó de 2 a 20 neuronas en la primera capa oculta, mientras que para la Figura 3. ***26*** se incrementó de 5 a 20 neuronas en la segunda capa oculta.

Finalmente se seleccionó una tasa de aprendizaje de 1e-7 para la configuración de estudio 2, obteniendo resultados favorables en la predicción de datos como se muestra en la Figura 3.17.

El número de iteraciones o epoch que debe realizar de red neuronal se estableció a partir de las Figura 3.27, Figura 3.28, Figura 3.29 y Figura 3.30 que pertenecen a las configuraciones de estudio 2,4,6,8 respectivamente, para las cuales se observa como su MSE varía en función de su epoch.

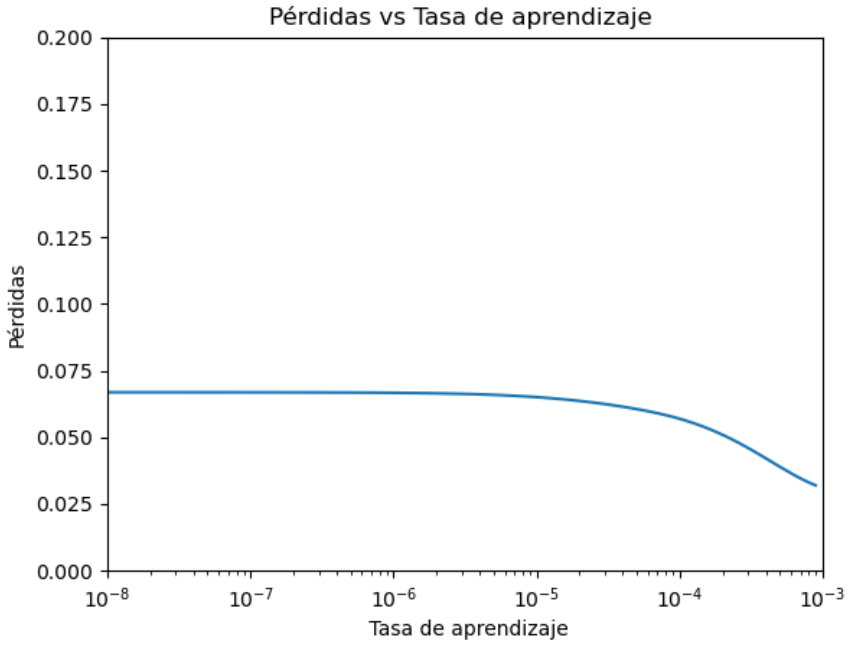


Figura 3. 24. Pérdida vs Tasa de aprendizaje para la configuración de estudio 2

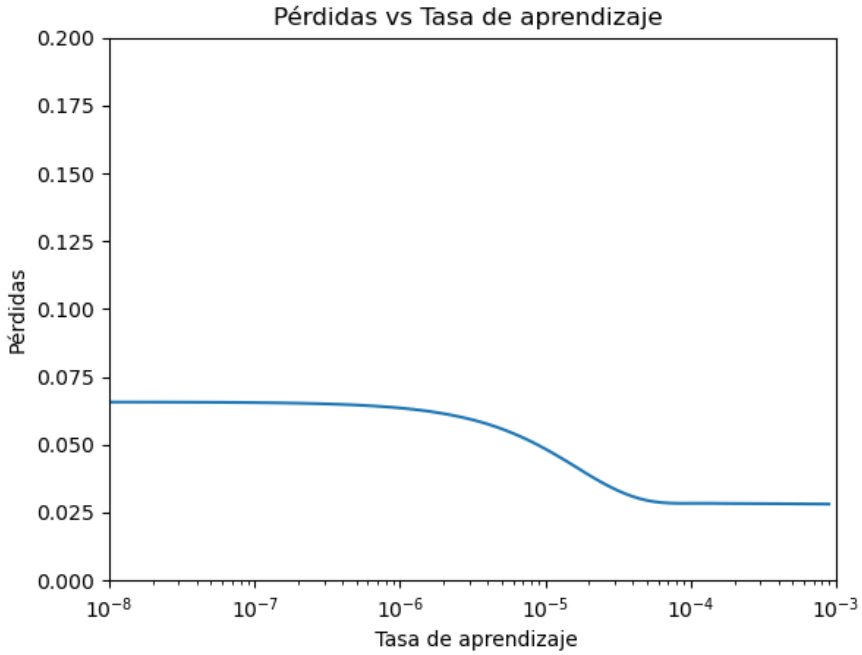


Figura 3. 25. Pérdida vs Tasa de aprendizaje para la configuración de estudio 2 modificada

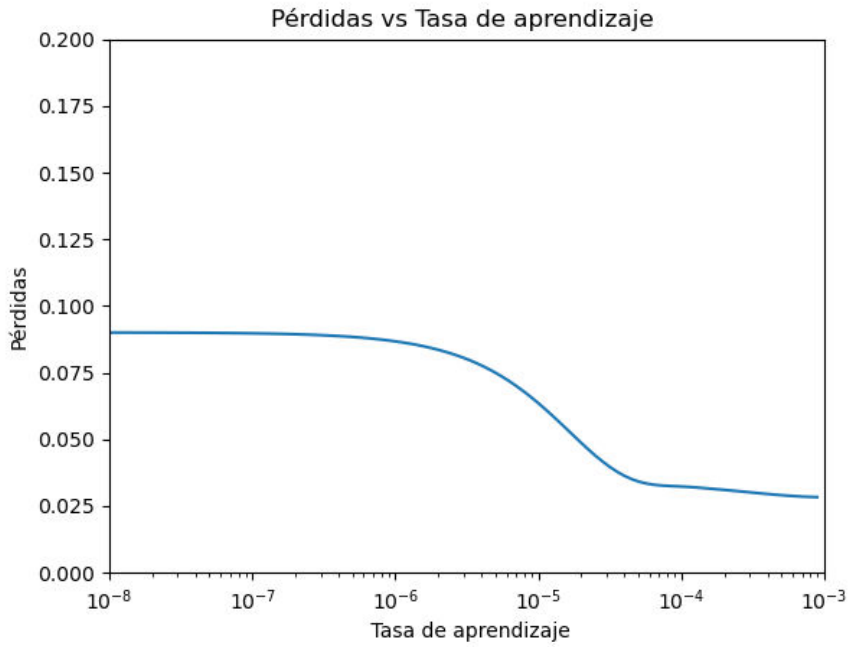


Figura 3. 26. Pérdida vs Tasa de aprendizaje para la configuración de estudio 2 modificada

Al igual que la tasa de aprendizaje el MSE de la red neuronal depende del número de capas y neuronas, por esta razón las Figura 3.27, Figura 3.28, Figura 3.29 y Figura 3.30 tienen una diferente forma, para las cuales se seleccionaron 50, 100, 100 y 100 iteraciones respectivamente.

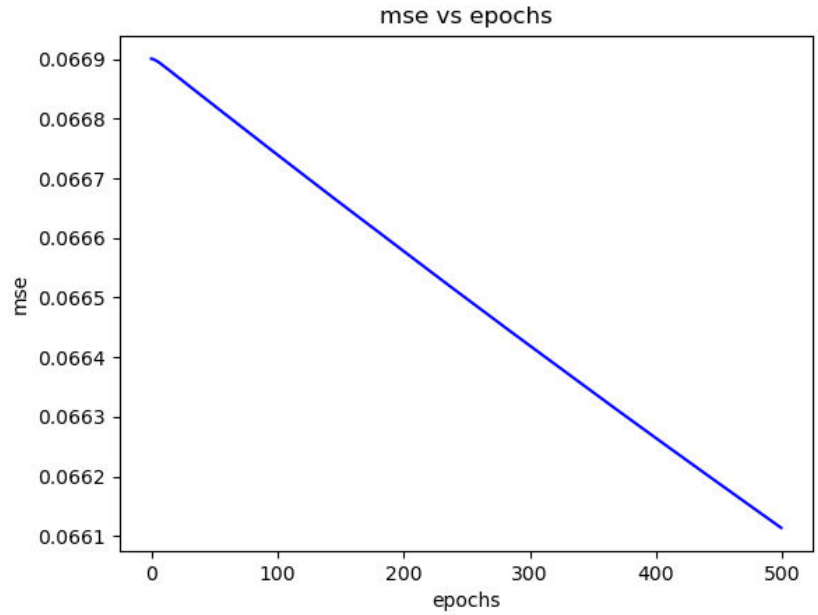


Figura 3.27. MSE vs epochs para la configuración de estudio 2

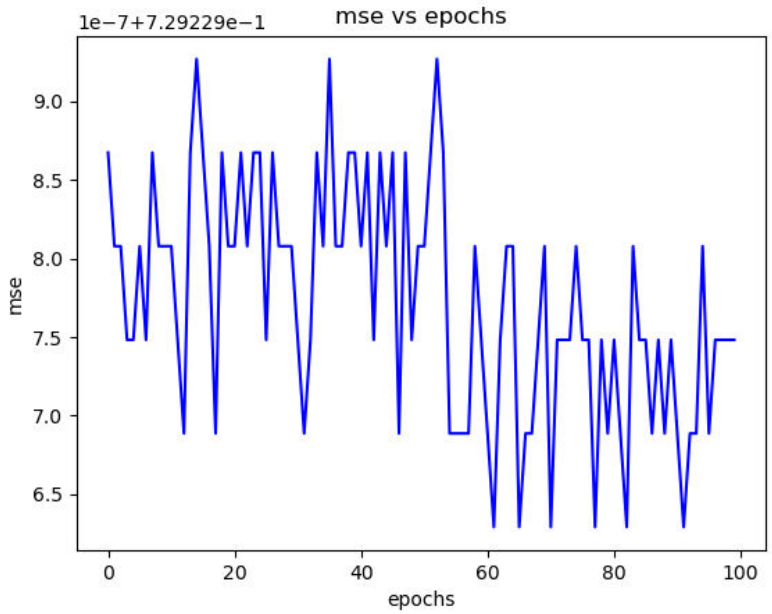


Figura 3.28. MSE vs epochs para la configuración de estudio 4

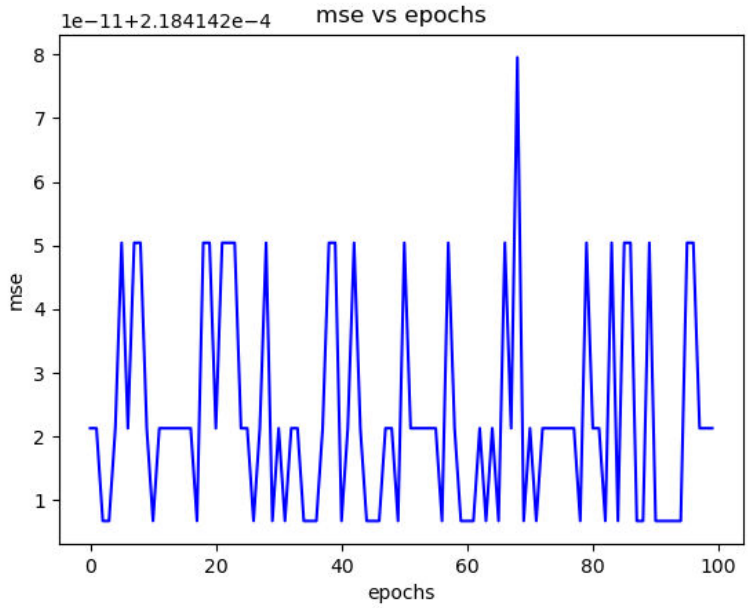


Figura 3.29. MSE vs epochs para la configuración de estudio 6

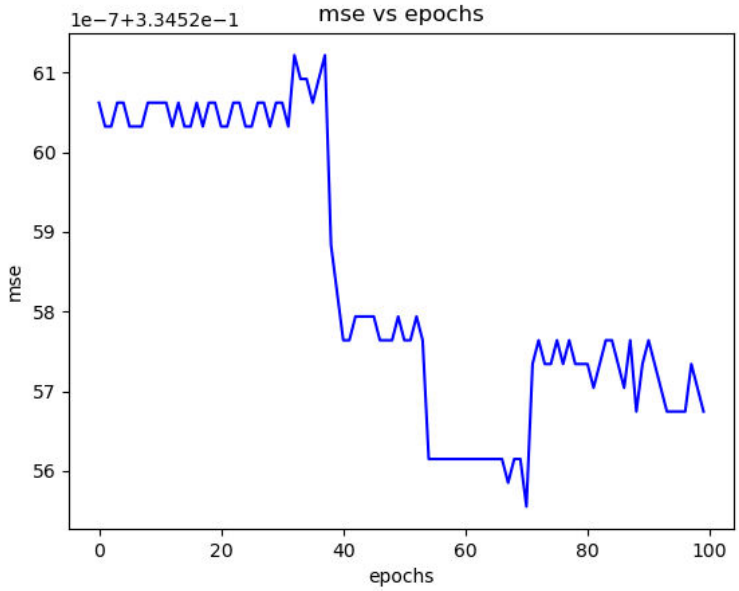


Figura 3.30. MSE vs epochs para la configuración de estudio 8

3.6. DISCUSIÓN DE RESULTADOS

En la sección 3.2, 3.3 y 3.4 se ha visualizado los resultados obtenidos de la basa de datos de eventos de falla, RNA y datos no normalizados, y RNA y datos normalizados, respectivamente. En todos los casos de RNA y datos normalizados su índice MSE es menor en comparación del uso de datos no normalizados.

La elección de la configuración óptima de RNA es en base al indicador MSE que presente menor valor y que su gráfica se ajuste de manera adecuada a los valores reales, los mismos que son las configuraciones de estudio 2,4,6 y 8. Donde cada una de estas configuraciones tiene un dato de entrada diferente, distancia de falla, fecha de falla, línea en falla y tipo de falla respectivamente.

Mientras que en la interfaz visual se logra apreciar las gráficas y datos de las líneas de transmisión además de las predicciones y su correspondiente gráfica, pudiendo ser ubicadas mediando doble clic en la celda de número de falla.

4. CONCLUSIONES Y RECOMENDACIONES

4.1. CONCLUSIONES

- A través de la investigación bibliográfica sobre el método de Montecarlo se estableció los pasos que debe seguir la metodología para crear la base de datos de fallas tomando información de las L/T del sistema de prueba IEEE 39 barras.

- Se desarrolló una base de datos que almacena eventos de falla en líneas de transmisión basada en el método de Simulación de Montecarlo, esta cuenta con campos para: distancia de falla, fecha de falla, L/T en falla, y tipo de falla. Se consideró que, al modificar el valor de la semilla para iniciar con la generación de números aleatorios la base datos cambiará completamente y esto a su vez influirá en la RNA ya que se deberá ser entrenada nuevamente.

- Al no contar con archivos de operación y número de fallas registradas, se realizó la estimación de las tasas de falla mediante el método de distribución Chi-Cuadrado, tomándose en cuenta que:

    - Las fallas llegan una a la vez.
    - El número de fallas que llegan en un intervalo de tiempo no afecta el número de fallas durante otro intervalo de tiempo.
    - Las fallas son independientes entre sí.
    - La tasa de falla es constante.
    - El tiempo de reparación de los componentes son despreciables en comparación con la tasa de falla.

- Las redes neuronales no necesitan de un modelo matemático o curva teórica para realizar comparaciones, característica que es de gran ayuda en casos donde la serie de datos no tiene un comportamiento conocido o son datos de origen aleatorio. Esto se evidencia al usar números pseudoaleatorios en donde la red neuronal busca patrones entre los datos ajustándolos mediante los pesos para finalmente dar como resultado un pronóstico eficiente. Para este trabajo de titulación al usar números pseudoaleatorios la RNA fue la herramienta adecuada para lograr la predicción de fallas.

- La normalización de los datos de entrada a la red neuronal artificial fue de suma importancia, ya que se conservan las características de los datos originales y permitió usar menos capas ocultas, y menos neuronas en estas capas, dando como resultado un bajo error y mejor precisión en la predicción de los datos.

- Al incrementar el número de capas ocultas y números de neuronas en la red artificial sin ser evaluadas produjo un sobre entrenamiento, afectando a los resultados y provocando un incremento de error en la precisión de las predicciones. La Tabla 3.4 muestra 5 capas ocultas teniendo un resultado deficiente en la predicción de las fallas, por otra parte, el incremento leve de neuronas y capas produce un mejor resultado en la predicción de las fallas como se muestra en la Figura 3.15.

- Se determinó que las RNAs que mejor se comportan para la predicción de fallas son las configuraciones de estudio 2,4,6,8, disminuyendo el error, lo que demuestra que la red neuronal puede usarse para pronosticar eventos de fallas en líneas de transmisión, ya que al observar las gráficas de las configuraciones mencionadas los datos pronosticados se ajustan de manera adecuada a los datos reales.

- Se implementó una interfaz para la visualización de los datos generados en el método de Montecarlo y los resultados de cada RNA, tanto gráficamente como numéricamente. Esto se evidencia en la Figura 3.5, Figura 3.6, Figura 3.7 y Figura 3. 8.

- Se obtuvo la gráfica de perdidas vs tasa de aprendizaje de una red neuronal la cual depende de su número de capas ocultas, así como por sus neuronas, mientras que su MSE depende de la tasa de aprendizaje y de su número de iteraciones como se muestra en la sección 3.5.

- Se implementó una red neuronal artificial, en la cual ingresan datos de distancia de falla, fecha de falla, línea en falla y tipo de falla. Sin embargo, para futuras investigaciones se podrían ingresar datos de voltajes, corrientes y potencias de los eventos de fallas con el fin de tener una mejor precisión en la predicción de fallas.

4.2. RECOMENDACIONES

- Se recomienda que el número semilla para empezar la generación de números aleatorios distribuidos uniformemente, para la base de datos y RNA sean el mismo valor ya que se consigue un mejor ajuste de predicción.

- Se recomienda realizar más casos de estudio para lograr un modelo de predicción con un valor de error menor al encontrado en el presente proyecto de titulación.

- Se recomienda obtener datos reales de fallas de empresas eléctricas para el desarrollo de un modelo de predicción, ya que evidenciaría la gran ayuda que pueden prestar las redes neuronales artificiales en el día a día de la empresa eléctrica, además recodar

que los datos tienen diferentes patrones por lo que se debe buscar y entrenar cada modelo de RNA.

- Se recomienda no colocar demasiadas capas ocultas y número de neuronas en las capas ocultas, ya que incrementa el tiempo de procesamiento del computador.

5. REFERENCIAS BIBLIOGRÁFICAS

[1] E. B. H. Leija, Localización de fallas en líneas de transmisión, Nuevo León: Universidad Autónoma de Nuevo León, Facultad de Ingeniería Mecánica y Eléctrica, 2014.

[2] M. I. Senlin Zhang, M. L. S. M. I. Yixing Wang y M. I. Zhejing Bao, “Data-based Line Trip Fault Prediction in Power Systems Using LSTM Networks and SVM,” IEEE, p. 11, 2017.

[3] D. T. Radmer, M. I. Paul A. Kuntz, M. I. Richard D. Christie, F. I. Subrahmanyam S. Venkata y M. I. Robert H. Fletcher, “Predicting Vegetation-Related Failure Rates for Overhead Distribution Feeders,” IEEE TRANSACTIONS ON POWER DELIVERY,, vol. VOL. 17, nº NO. 4, p. 6, OCTOBER 2002.

[4] C. M. Bishop, Neural Networks for Pattern Recognition, Oxford: Clarendon Press, 1995.

[5] V. M. I. G. Mohammed Mahdi, “Artificial Neural Network Based Algorithm for Early Prediction of Transient Stability Using Wide Area Measurements,” IEEE , p. 5, 2017.

[6] J. Doyle y L. E. Zaffanella, Transmission Line Referece Book 345 kV and Above Second Edition, Chapter 1, Palo Alto, Calofornia: J.J. LaForest, 1982.

[7] D. A. G. RODAS, DISEÑO E IMPLEMENTACION DE UNA HERRAMIENTA DE SOFTWARE PARA EL ANÁLISIS DE CONFIABILIDAD DE SISTEMAS ELÉCTRICOS DE POTENCIA BASADO EN EL MÉTODO DE SIMULACIÓN DE MONTECARLO, Quito: ESCUELA POLITECNICA NACIONAL, 2016.

[8] M. T. S. Escobar, ANÁLISIS ESTADÍSTICO DEL AÑO 2002 AL 2007 DE FALLAS EN LÍNEAS DE TRANSMISIÓN DE 69KV, 138KV Y 230KV DE LA EMPRESA DE TRANSPORTE Y CONTROL DE ENERGÍA ELÉCTRICA DEL INDE, Guatemala: Universidad de san Carlos de Guatemala, 2010.

[9] W. Vargas, “Implementación de Algoritmos de Detección, Clasificación y Localización de Fallas en Líneas de Transmisión Utilizando Medidas Fasoriales Sincronizadas.,” Energía, nº 13, 2017.

[10] I. Olalla Merino Wrom y I. Orejuela Luna Victor, “MEJORAMIENTO DE INDICES DE CONFIBILIDAD MEDIANTE LA OPTIMIZACION DE RECURSOS EN REDES DE DISTRIBUCION,” JIEE, vol. 19, p. 10, 2005.

[11] C. J. ZAPATA, “APLICACIONES DEL PROCESO DE POISSON EN CONFIABILIDAD,” SCIENTIA ET TECHNICA , vol. 20, p. 6, Octubre 2002.

[12] R. Billinton y R. N. Allan, Realiability Evaluation of Engineering Systems Concepts and Techniques, New York: Plenum Press, 1992.

[13] IEEE Recommended Practice for the Design of Reliable Industrial and Commercial Power Systems, IEEE Std 493™-2007 (Revision of IEEE Std 493-1997).

[14] C. J. ZAPATA, “ESTIMACIÓN DE TASAS DE FALLAS DE COMPONENTES EN CASOS DE AUSENCIA DE DATOS O CANTIDADES LIMITADAS DE DATOS,” Scientia Et Technica, vol. vol. XI, nº núm. 27, pp. pp: 13-18, abril, 2005.

[15] V. Y, “Probability and random processes for electrical engineers, McGraw Hill, 1998.

[16] C. J. ZAPATA, ANALISIS PROBABILISTICO Y SIMULACION, PEREIRA: UNIVERSIDAD TECNOLÓGICA DE PEREIRA, 2010.

[17] R.-. L. T. LLAMES, TÉCNICAS AVANZADAS DE PREDICCIÓN PARA BIG DATA EN EL CONTEXTO DE SMART CITIES, SEVILLA: UNIVERSIDAD PABLO DE OLAVIDE, DICIEMBRE 2018.

[18] P. Bunnoon, “Mid-Term Load Forecasting Based on Neural Network Algorithm: a Comparison of Models,” International Journal of Computer and Electrical Engineering,, vol. Vol. 3, nº No. 4,, pp. pp. 600-605, August 2011.

[19] J. A. RAMÍREZ, H. O. SARMIENTO y J. M. L.-L. , “Diagnóstico de fallas en procesos industriales mediante inteligencia artificial,” ESPACIOS, vol. Vol. 39 , nº (Nº 24) , p. 12, 2018.

[20] P. Bunnoon, K. Chalermyanont y C. Limsakul, “A Computing Model of Artificial Intelligent : a state-of-the-art- survey for the researcher Approaches to Mid-term Load Forecasting,” IACSIT International Journal of Engineering and Technology, Vols. %1 de %2Vol. 2,, nº No.1, pp. pp.94-100, February, 2010.

[21] C. M. Bishop, Neural Networks for Pattern Recognition, Great Britain: OXFORD UNIVERSITY PRESS, 2005.

[22] MetaQuotes, “Redes Neuronales: de la Teoría a la Práctica,” 2014. [En línea]. Available: https://www.mql5.com/es/articles/497. [Último acceso: 5 11 2020].

[23] U. I, “Guia metodológica para la selección de técnicas de depuración de datos”, trabajo, MEDELLIN: Univ. Nacional de Colombia, 2010.

[24] S. E y T. V, “ “Tratamiento de valores perdidos y atípicos en la aplicación del Modelo Estadístico de Medición de Impacto en un estudio de 90 fincas lecheras en la provincia de Pastaza, Ecuador”,,” CUBA, vol. vol.48, nº N 4, pp. pp. 333-336, ABRIL 2014.

[25] S. Mukhopadhyay, Advanced Data Analytics Using Python With Machine Learning, Deep Learning and NLP Examples, Kolkata, West Bengal, India: apress, 2018.

[26] L. A. F. Jimenez, Modelos avanzados para la prediccion a corto plazo de la produccion electrica en parques eolicos, Tesis Doctoral, Logroño: Universidad de la Rioja , Junio 2007.

[27] C. Albon, Python Machine Learning Cookbok PRACTICAL SOLUTIONS FROM PREPOCESSING TO DEEP LEARNING, United States of America: O'REILLY, Abril 2018.

[28] D. M, Redes Neuronales: Conceptos Basicos y Apliaciones., GIAIQ, Marzo 2001.

[29] P. P. Cruz, Inteligencia Artificial con aplicaciones a la ingenieria, México D.F: Alfaomega, 2010.

[30] J. R. Hilera y V. J. Martinez, REDES NEURONBALES ARTIFICIALES. Fundamentos, modelos y aplicaciones, MADRID: RA-MA Editrial, Enero 1995.

[31] R. Q. A. Requena, J. Bolarín, A. Vázquez, A. Bastida, J. Zúñiga y L. Tomás., "Equivalencia entre redes artificiales y biológicas," [En línea]. Available: https://www.um.es/LEQ/Atmosferas/Ch-VI-3/C63s4p3.htm. [Último acceso: 4 12 2020].

[32] J. Freeman y D. Skapura, Redes neuronales Algoritmos, aplicaciones y técnicas de, USA: Addison-Wesley Iberoamericana, 1993.

[33] planet.es.python.org, "Funciones de activación para un perceptron," 14 02 2018. [En línea]. Available: http://planet.es.python.org/index-2018-02-14.html. [Último acceso: 10 12 2020].

[34] D. Calvo, "Clasificación de redes neuronales artificiales," 13 Julio 2017. [En línea]. Available: https://www.diegocalvo.es/clasificacion-de-redes-neuronales-artificiales/. [Último acceso: 9 11 2020].

[35] J. Brownlee, Deep Learning for Time Series Forecasting Predict the Future with MLPs, CNNs and LSTMs in Python, 2019.

[36] C. J. ZAPATA, L. C. PIÑEROS y D. A. CASTAÑO, "EL MÉTODO DE SIMULACIÓN DE MONTECARLO EN ESTUDIOS DE CONFIABILIDAD DE SISTEMAS DE DISTRIBUCIÓN DE ENERGÍA ELÉCTRICA," Scientia et Technica Año X, nº No 24, pp. pp. 55-60, Mayo 2004.

[37] deeplearning.ai, "Sequences, Time Series and Prediction," [En línea]. Available: https://www.coursera.org/learn/tensorflow-sequences-time-series-and-prediction/home/welcome?. [Último acceso: 10 12 2020].

[38] F. Berzal, REDES NEURONALES Y DEEP LEARNING, GRANADA, 2018.

[39] N. N. Community, "What is NumPy?," 18 02 2020. [En línea]. Available: https://numpy.org/doc/1.17/user/whatisnumpy.html. [Último acceso: 11 12 2020].

[40] S. O. contributors, "LEARNING pandas," [En línea]. Available: https://riptutorial.com/Download/pandas.pdf. [Último acceso: 11 12 2020].

[41] J. Hunter, D. Dale, E. Firing y M. Droettboom, Matplotlib Release 2.0.2, 2017.

[42] A. Géron, Hands-On Machine Learning with Scikit-Learn & TensorFlow CONCEPTS, TOOLS, AND TECHNIQUES TO BUILD INTELLIGENT SYSTEMS, United States of America: O'REILLY, 2017 .

[43] Keras, "Keras Doocumentation," 2020. [En línea]. Available: https://keras.io/. [Último acceso: 11 12 2020].

[44] B. Harwani, Qt5 Python GUI Programming Cookbook, BIRMINGHAM - MUMBAI: Packt Publishing, 2018.

[45] L. Carvajal, Metodología de la Investgación Científica. Curso general y aplicado, 28 ed., Santiago de Cali: U.S.C., 2006, p. 139.

ANEXOS

ANEXO A. Interfaz gráfica en leguaje de programación Python para el modelo de predicción de fallas.

ANEXO B. Generación de la base de datos de eventos de falla em leguaje de programación Python para el modelo de predicción de fallas.

ANEXO C. Scripts de RNA y datos no normalizados en leguaje de programación Python para el modelo de predicción de fallas.

ANEXO D. Scripts de RNA y datos normalizados en leguaje de programación Python para el modelo de predicción de fallas