



**UNIVERSIDAD TECNOLÓGICA ISRAEL**  
**ESCUELA DE POSGRADOS “ESPOG”**

**MAESTRÍA EN ELECTRÓNICA Y AUTOMATIZACIÓN**  
*Resolución: RPC-SO-09-No.265-2021*

**PROYECTO DE TITULACIÓN EN OPCIÓN AL GRADO DE MAGISTER**

<b>Título del proyecto:</b>
Detección de fallas en un motor usando el método de machine learning
<b>Línea de Investigación:</b>
Ciencias de la ingeniería aplicadas a la producción, sociedad y desarrollo sustentable
<b>Campo amplio de conocimiento:</b>
Ingeniería, industria y construcción
<b>Autor/a:</b>
Jeannette Virginia Sánchez Cocha
<b>Tutores:</b>
PhD Maryory Urdaneta Herrera Mg. Wilmer Fabian Albarracín Guarochico

**Quito – Ecuador**

**2024**

## APROBACIÓN DEL TUTOR



Yo, **PhD Maryory Urdaneta** con C.I: **175931612-6** en mi calidad de Tutor del proyecto de investigación titulado: **Detección de fallas en un motor usando el método de machine learning.**

Elaborado por: **Jeannette Virginia Sánchez Cocha**, de C.I: **091537563-8**, estudiante de la Maestría: **ELECTRÓNICA Y AUTOMATIZACIÓN**, de la **UNIVERSIDAD TECNOLÓGICA ISRAEL (UISRAEL)**, como parte de los requisitos sustanciales con fines de obtener el Título de Magister, me permito declarar que luego de haber orientado, analizado y revisado el trabajo de titulación, lo apruebo en todas sus partes.

Quito D.M., 30 de agosto de 2024



---

**Firma**

## APROBACIÓN DEL TUTOR



Yo, **Mg. Wilmer Fabian Albarracín Guarochico** con C.I: **171334115-2** en mi calidad de Tutor del proyecto de investigación titulado: **Detección de Fallas en un motor usando el método de machine learning.**

Elaborado por: **Jeannette Virginia Sánchez Cocha**, de C.I: **091537563-8**, estudiante de la Maestría: **ELECTRÓNICA Y AUTOMATIZACIÓN** de la **UNIVERSIDAD TECNOLÓGICA ISRAEL (UISRAEL)**, como parte de los requisitos sustanciales con fines de obtener el Título de Magister, me permito declarar que luego de haber orientado, analizado y revisado el trabajo de titulación, lo apruebo en todas sus partes.

Quito D.M., 30 de agosto de 2024



---

**Firma**

## DECLARACIÓN DE AUTORIZACIÓN POR PARTE DEL ESTUDIANTE



Yo, **Jeannette Virginia Sánchez Cocha** con C.I: **091537563-8**, autor/a del proyecto de titulación denominado: **Detección de fallas en un motor usando el método de machine learning**. Previo a la obtención del título de Magister en **ELECTRÓNICA Y AUTOMATIZACIÓN**.

1. Declaro tener pleno conocimiento de la obligación que tienen las instituciones de educación superior, de conformidad con el Artículo 144 de la Ley Orgánica de Educación Superior, de entregar el respectivo trabajo de titulación para que sea integrado al Sistema Nacional de Información de la Educación Superior del Ecuador para su difusión pública respetando los derechos de autor.
2. Manifiesto mi voluntad de ceder a la Universidad Tecnológica Israel los derechos patrimoniales consagrados en la Ley de Propiedad Intelectual del Ecuador, artículos 4, 5 y 6, en calidad de autor@ del trabajo de titulación, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente. En concordancia suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital como parte del acervo bibliográfico de la Universidad Tecnológica Israel.
3. Autorizo a la SENESCYT a tener una copia del referido trabajo de titulación, con el propósito de generar un repositorio que democratice la información, respetando las políticas de prosperidad intelectual vigentes.

Quito D.M., 30 de agosto de 2024

---

Firma

## Tabla de contenidos

APROBACIÓN DEL TUTOR	ii
DECLARACIÓN DE AUTORIZACIÓN POR PARTE DEL ESTUDIANTE	iii
INFORMACIÓN GENERAL	1
Contextualización del tema	1
Problema de investigación	1
Objetivo general	1
Objetivos específicos	1
Vinculación con la sociedad y beneficiarios directos:	1
CAPÍTULO I: DESCRIPCIÓN DEL PROYECTO	2
1.1. Contextualización general del estado del arte	2
1.2. Proceso investigativo metodológico	2
1.3. Análisis de resultados	2
CAPÍTULO II: PROPUESTA	3
2.1. Fundamentos teóricos aplicados	3
2.2. Descripción de la propuesta	3
2.3. Validación de la propuesta	3
2.4. Matriz de articulación de la propuesta	4
CONCLUSIONES	5
RECOMENDACIONES	6
BIBLIOGRAFÍA	7
ANEXOS	8

## Índice de tablas

Tabla 1 <i>Parámetros y Entradas del Motor DC</i>	13
Tabla 2 <i>Datos de la red</i>	17
Tabla 3 <i>Comparación entre modelos de red</i>	17
Tabla 4 <i>Matriz de articulación</i>	22
Tabla 5 <i>Comparación de características</i>	24
Tabla 6 <i>Comparación de modelos Machine Learning</i>	24

## Índice de figuras

Figura 1 <i>Diagrama de motor DC</i>	9
Figura 2 <i>Proceso de Sistema de Diagnóstico</i>	10
Figura 3 <i>Programa Simulink</i>	12
Figura 4 <i>Programa MATLAB</i>	12
Figura 5 <i>Arquitectura de Red Neuronal</i>	13
Figura 6 <i>Modelo Motor DC</i>	14
Figura 7 <i>Generador de Fallas</i>	14
Figura 8 <i>Función de Identificador de Fallas</i>	15
Figura 9 <i>Predictores y Resultados de la Red</i>	15
Figura 10 <i>Configuración del Entrenamiento</i>	16
Figura 11 <i>Red Neuronal en Quartus</i>	17
Figura 12 <i>Vectores de Prueba - Matlab</i>	18
Figura 13 <i>Escenario de prueba 1 - Simulink</i>	18
Figura 14 <i>Escenario de prueba 2 - Simulink</i>	19
Figura 15 <i>Escenario de prueba 3 - Simulink</i>	19
Figura 16 <i>Vector de prueba y escenario de prueba 1 - Quartus</i>	20
Figura 17 <i>Vector y escenario de prueba 2 - Quartus</i>	20
Figura 18 <i>Vector y escenario de prueba 3 - Quartus</i>	21
Figura 19 <i>Performance Levenberg-Manquardt</i>	25
Figura 20 <i>Performance Bayesian Regularization</i>	25
Figura 21 <i>Performance Scaled Conjugate Gradient</i>	26
Figura 22 <i>Performance Levenberg-Manquardt (elegido)</i>	26
Figura 23 <i>Error Histogram</i>	27
Figura 24 <i>Regression</i>	27
Figura 25 <i>Resultados del entrenamiento</i>	28
Figura 26 <i>Red Neuronal para el Sistema</i>	28
Figura 27 <i>Gráfico im -Healthy-</i>	29
Figura 28 <i>Gráfico wm -Healthy-</i>	29
Figura 29 <i>Gráfico de Señal de Falla de Offset</i>	30
Figura 30 <i>Gráfico de im -Rm-Fault-</i>	30
Figura 31 <i>Gráfico de wm -Rm-Fault-</i>	31
Figura 32 <i>Parámetros de la red neuronal</i>	31
Figura 33 <i>Bloque de Red neuronal -Quartus-</i>	32

## INFORMACIÓN GENERAL

### Contextualización del tema

Las técnicas para el análisis de motores constituyen un factor importante para el mantenimiento predictivo, conocer los principios y las diferencias entre ellas pueden asegurar la confiabilidad y disponibilidad del motor dentro del sistema en que trabajen.

Los motores de corriente continua, llamados motores dc, se emplean cuando se requiere controlar la velocidad fina y/o alto par en baja velocidad, como son: tracción eléctrica de vehículos; motores lineales en ferrocarriles; maquinaria industrial en trefiladoras, bobinadoras, elevadoras y extractoras; robótica y automatismo; medicina en equipos de bombeo y resonancia; aeroespacial en satélites y control de vuelo (International Couplings, 2016).

”La técnica de machine learning utilizada para el mantenimiento predictivo de motores, transforma las operaciones de mantenimiento de reactivo a proactivo” (Gosalbez, 2024).

Este trabajo muestra la técnica de machine learning para diagnosticar las fallas presentadas en motores dc. AMFEC (2017) indica que las fallas que comúnmente se presentan en este tipo de motores son: desequilibrio en el circuito de alimentación, desequilibrio de corriente o daños en el bobinado y perturbaciones por variaciones de carga. El comportamiento del motor frente a estas fallas es el bajo rendimiento que se traduce en pérdidas o desgaste mecánico, reducción de la fuerza electromotriz, giro a mayor o menor velocidad, transitorios en el control, aumento de la temperatura, entre otros.

Un tipo de motor fué considerado para el estudio, se trata de un motor de configuración tipo serie de corriente continua, para el que se pretende extender su ciclo útil, al realizar el chequeo de las variables que consiguen perjudicar su desempeño. Dichas variables pueden ser identificadas y tratadas previamente por medio de acciones correctivas, logrando con esto reducir costos de reparación y disminuir tiempos de inactividad del motor. Debido a estas ventajas, se muestra cómo es utilizado el machine learning (ML) en este tipo de aplicación y cómo puede ser implementado como Edge ML y más específicamente cómo Embedded ML al utilizar algoritmos para la selección y extracción de las características que permiten obtener la precisión requerida para la detección de la falla del motor.



## **Problema de investigación**

Actualmente los motores eléctricos cuentan con una técnica de diagnóstico llamada mantenimiento predictivo que requiere la recogida y análisis de datos con el uso de machine learning, esta técnica como menciona Sardashti y Nazari (2023) “logra un rendimiento óptimo mientras mantiene una baja carga computacional” y según Bhatia y Vishwakarma (2023) “mejora las operaciones industriales al reducir los gastos de reparación, el tiempo de inactividad y la seguridad” pero requiere conocimiento estadístico y/o matemático.

Existe falta de modelos digitales de motores eléctricos en la industria, que puedan ser utilizados como referencia para los mantenimientos predictivos.

Muchas industrias no realizan el mantenimiento predictivo como menciona CORDIS (2021) porque ya de por sí “se destina un presupuesto considerable en el mantenimiento”.

Los motores necesitan ser monitoreados continuamente y se espera que no se produzcan paros no programados que ocasionan costos adicionales por la falta de mantenimiento.

## **Objetivo general**

Desarrollar una propuesta de detección de fallas en el motor usando el método de machine learning.

## **Objetivos específicos**

- Contextualizar las bases teóricas de los sistemas de detección de fallas con el uso de machine learning.
- Diseñar el sistema de detección de fallas basado en machine learning.
- Analizar los parámetros estadísticos y de clasificación de machine learning.
- Validar el modelo de machine learning.

## **Vinculación con la sociedad y beneficiarios directos:**

Aplicar el mantenimiento predictivo permite reducir gastos de mantenimiento como: materia prima, personal asignado, repuestos, y que estos valores puedan ser empleados en mejoras o adquisición de materiales para el fortalecimiento de la industria y reducir la huella de contaminación al medioambiente.

Este modelo de diagnóstico permite contribuir al área industrial, sobre el uso de herramientas digitales empleadas en trabajos técnicos.

## CAPÍTULO I: DESCRIPCIÓN DEL PROYECTO

### 1.1. Contextualización general del estado del arte

El diagnóstico adecuado puede ser la diferencia en la extensión del tiempo de funcionamiento y el correcto desempeño del motor. Existen diversas técnicas de diagnóstico, de estas destaca el mantenimiento predictivo.

La investigación se centró en la técnica del machine learning para predecir las fallas de un motor dc. Se utilizó el modelo matemático del motor dc y se recopilaron datos generados a través de simulación.

La fuente de información se basa en videos y documentos de revistas y tesis obtenidas en línea de índole científico y técnico de acceso libre y por suscripción.

Se ha establecido la búsqueda de información con la ayuda de palabras clave como son los términos: motor, diagnóstico, fallas y machine learning.

De la información recopilada se puede observar la ruta de trabajo que los autores realizaron para el desarrollo del estudio, se puede realizar comparaciones del estudio entre ellos y elegir la más adecuada para el análisis.

Las investigaciones van dirigidas a encontrar el método de diagnóstico más adecuado para un determinado tipo de motor. Existe la tendencia de usar redes neuronales sean estas artificiales o de convolución para la detección de fallas. Los resultados se han aplicado a motores de inducción, síncrono y corriente directa. Y se pueden transferir a una computadora para generar referencias de diagnóstico para diversos casos de prueba.

Machine learning es la ciencia que desarrolla algoritmos y modelos estadísticos para procesar gran cantidad de datos para identificar patrones (Amazon).

El machine learning es utilizado para el mantenimiento predictivo.

El mantenimiento predictivo busca anticipar el fallo para evitar daños en el motor.

Se deberá tener conocimiento sobre machine learning, elaboración de modelos y uso de software de análisis de datos para resolver el problema.

Existe un autor que propone el “análisis estadístico y ANN para la clasificación de las fallas del motor de inducción, además provee una implementación en FPGA para el monitoreo de la condición en tiempo real”(Cabal-Yépez et al., 2014, pp. 1657-1672).

Otro autor se concentra en “la evaluación de fallas en las primeras etapas, y con la utilización de ANN, mostró resultados de correlación cercano a 1 para la identificación del modelo y clasificación de fallas”(Restrepo et al., 2017).

“Según el estudio del sistema de diagnóstico de fallas que detecta tempranamente las condiciones de operación y fallas en motores de inducción; el sistema se fundamenta en un modelo CNN, que produjo resultados de diagnóstico con una precisión de 98% y 100 % para tres estados del motor“(Jong-Hyun et al., 2019).

“Existe otro método para detectar fallas en la bobina del estator de motores de inducción, el método se basa en dos modelos de ANN, que presentaron dificultad para predecir fallas de etapas iniciales pero obtuvo mejores resultados de 100% de éxito para fallas de mayor severidad“(Vicente et al., 2020).

“Las técnicas de machine learning aplicadas al área industrial determinan que esta es una estrategia de predicción que debe trabajar en conjunto con el mantenimiento correctivo de prevención“(Castro, 2022).

Se emplearon instrumentos para exploración en datos y se comparó el resultado de la precisión al repetir las prácticas de análisis, que permitió elegir el mejor modelo de detección de fallas.

Se utilizó el aprendizaje supervisado porque se analizan las entradas y salidas del modelo del motor.

La mayoría de estudios de detección de fallas que existen realizados por diversos autores, no se realizaron directamente en motores dc, sino en el driver o el circuito de control del driver.

## **1.2. Proceso investigativo metodológico**

El enfoque es del tipo cuantitativo, porque recoge las señales que evalúan el rendimiento del motor y realiza el análisis que requiere la detección de dos tipos de casos con o sin falla, permitiendo identificar el tipo de falla para que se pueda proceder a dar solución al mismo.

Los datos generados del simulador se registran para su posterior análisis.

La información recogida contribuye con la elección del criterio más adecuado para solucionar la problemática actual.

Para el análisis, de los grupos de muestras, corresponden 80% muestras para entrenamiento, 10% de muestras para validación y 10% muestras para prueba. Se consideran 9

estados del motor: normal, falla 1 a falla 8, lo que permite tener una combinación de un valor total de 257 muestras.

El muestreo es de tipo sistemático porque los datos se ordenan en una lista y serán seleccionados por grupos para entrenar la red mientras se varía la configuración del número de neuronas y el tipo de función utilizada en las capas internas.

El método de trabajo consistió de cuatro etapas: el análisis de las herramientas a utilizar, caracterización del objeto de estudio en conjunto con el diseño e implementación del banco de trabajo con las herramientas seleccionadas, creación de escenarios para los experimentos y evaluación de los escenarios.

## CAPÍTULO II: PROPUESTA

### 2.1 Fundamentos teóricos aplicados

El machine learning es la ciencia que utiliza algoritmos estadísticos para la búsqueda de patrones usando un modelo que aprende y realiza ajustes automáticamente, a partir de datos.

Morales (2023) indica que una máquina realiza conocimiento nuevo a medida que se le expone a mayor información.

Freire (2023) menciona que un sistema basado en machine learning se le enseña a tomar decisión con la incorporación de nuevos datos.

Por ello es comprensible que el machine learning sea muy utilizado para automatizar procesos y disminuir la incertidumbre.

La detección e identificación de fallas es el componente principal en varios sistemas automatizados de gestión de operaciones. Cuando se observa un síntoma es necesario detectar y separar fallas. Las técnicas de detección se utilizan típicamente para una sola variable y las técnicas basadas en modelo para múltiples variables, siendo una de estas el machine learning, que pueden detectar eventos, tendencias, ser modelos de diagnóstico o clasificadores de fallas. Lei et al. (2020), comenta las aplicaciones de las teorías de machine learning como son ANN, SVM y DNN en el diagnóstico de fallas de máquinas. Sardashti y Nazari (2023) aprovechan las capacidades del machine learning para lograr un óptimo rendimiento en su método de detección de fallas de motores de corriente continua.

El tema específico que trata este trabajo fué la realización de un sistema para detectar fallas (FDI), basado en machine learning, para reducir fallas ante comportamientos defectuosos en motores dc.

Términos como falla, problema y causa de falla se relacionan pero no son iguales. Un problema se define como operación no óptima o producto fuera de las especificaciones. Causa de falla se asocia con procedimientos para reparar y Falla se considera como medida de ineficiencia.

A continuación se definen los conceptos principales y las bases teóricas relacionadas para el planteamiento del problema.

#### 2.1.1. Mantenimiento predictivo

Consiste de las acciones correctivas antes de que ocurra una falla. Primero se establece el estado del equipo y se estima cuándo realizar alguna acción para evitar fallas inesperadas.

En la industria el mantenimiento predictivo se aplica mayormente a las máquinas eléctricas. Los componentes que afectan el resultado del mantenimiento predictivo son los elementos fijos (estructura externa, base, etc.), elementos variables (estructura y materiales internos) y datos externos (condiciones climáticas).

### **2.1.2. Machine learning**

Como se mencionó anteriormente, el machine learning se utiliza en la detección de fallas, para ello utiliza algoritmos y posee una arquitectura. Existen cuatro tipos de algoritmos de machine learning para entrenar modelos:

- Aprendizaje por refuerzo: acciones, reglas y estados. Usa prueba y error
- Aprendizaje supervisado: datos (entrada, salida) donde la salida se etiqueta. Usa ejemplos
- Aprendizaje semi-supervisado: datos etiquetados y no etiquetados. Busca las propiedades
- Aprendizaje no supervisado: datos (entrada). Busca datos similares para agrupar

La estructura de machine learning comprende una red neuronal, compuesta de nodos, neuronas y conexiones arreglados en capas, las neuronas usan procesos imitando a las neuronas biológicas que trabajan en conjunto para identificar fenómenos, evaluarlos y llegar a conclusiones. Los pasos para construir aplicaciones de machine learning son:

- Seleccionar entradas y salidas
- Adquirir datos de entrenamiento de entradas/salidas
- Pre-procesar datos para normalizar si fuese necesario
- Entrenar la red
- Validar la red
- Aplicar la red
- Periódicamente re-entrenar para adaptación

El machine learning trabaja con el entrenamiento y puede contener errores y sesgos si existe información incorrecta producto del error humano o por los algoritmos.

El riesgo que posee el uso de machine learning es de proveer un resultado de correlación entre datos que no existe, debido a la complejidad del algoritmo y de los resultados. Así se producen los llamados:

-Caja Negra: no se conoce datos de entrenamiento, ni del algoritmo y ni del modelo

-Caja Blanca: se conoce el algoritmo y el modelo

-Caja Gris: parcialmente se conoce el algoritmo, modelo y datos

### **2.1.2.1 Arquitectura de Redes neuronales**

#### **2.1.2.1.1 Multicapa perceptron MLP**

Se trata de una arquitectura que posee asignación no lineal entre la entrada y la salida. Se tiene las capas llamadas: entrada, escondida y salida. El valor total de capas ocultas determina la profundidad de la red neuronal.

Las neuronas se conectan entre sí entre las capas. Cada neurona recibe una serie de entradas y calcula un valor de suma ponderada. La neurona se activará o no basándose en la salida de la función de activación, el cual toma todos los valores de entrada y los pesos junto con un sesgo (bias) y calcula un número. Este número es un valor entre -1 y 1 o 0 y 1, para la elegida función de activación.

Entre las funciones de activación más conocidas destacan:

-Sigmoid.-Traza la evolución del proceso en espacio de tiempo, consiguiendo bajar la tasa de error. Se usa en la regresión logística.

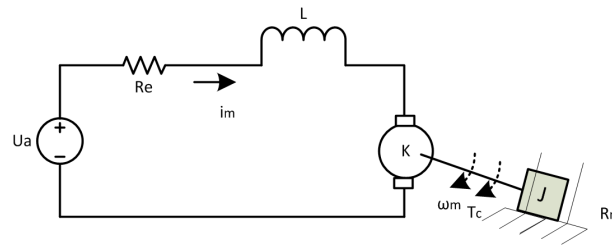
-Forward Back Propagation.- Trabaja en varias capas desde la entrada hasta la salida. Al llegar a la última capa, calcula el error y lo propaga desde la salida a la entrada, ajustando el peso de las neuronas.

### **2.1.3. Motor DC**

El motor transforma la energía eléctrica DC a mecánica. Existen cuatro tipos de motor de corriente directa (DC) clasificados según como se conecta su bobinado entre sí: Excitación independiente, Auto-excitación: En serie, Derivación o motor Shunt, Compound. El circuito equivalente ideal del motor DC (Figura 1) es:

**Figura 1**

*Diagrama de motor DC*



Cuyos parámetros y entradas son:

$U_a$ : Voltaje de entrada,  $R_e$ : Resistencia eléctrica,  $K_e$ : Constante eléctrica,  $L$ : Inductancia,  $T_c$ : Torque de la carga,  $R_m$ : Resistencia mecánica,  $K_m$ : Constante mecánica,  $J$ : Constante de Inercia

Los variables de funcionamiento del motor son  $i_m$  : Corriente y  $\omega_m$  : Velocidad. Las ecuaciones relacionadas son:

$$i_m = \frac{1}{L} * \int (U_a - R_e * i_m - K_e * \omega_m )$$

$$\omega_m = \frac{1}{J} * \int (K_m * i_m - R_m * \omega_m - T_c )$$

#### **2.1.4. Sistema embebido**

Es un sistema que combina hardware y software para realizar una tarea específica. Los componentes de un sistema embebido son: CPU, RAM, interfaces I/O. Existen tres categorías de sistemas embebidos: PLC, Microcontroladores, System on Chip.

##### **2.1.4.1 System On Chip (SOC)**

Es una tecnología integradora de varios componentes: los tradicionales (procesador, memoria, interfaz I/O) y los especiales (procesador gráfico, inalámbrico, etc.). Además puede correr sistemas operativos. Existen 2 tipos: ASIC, FPGA

##### **2.1.4.1.1 Compuerta de Arreglo Programable (FPGA)**

Es un semiconductor que se compone de lógica programable y conexiones que pueden unirse para realizar una función lógica. Se programan en lenguaje de descripción de hardware HDL.



## 2.2 Descripción de la propuesta

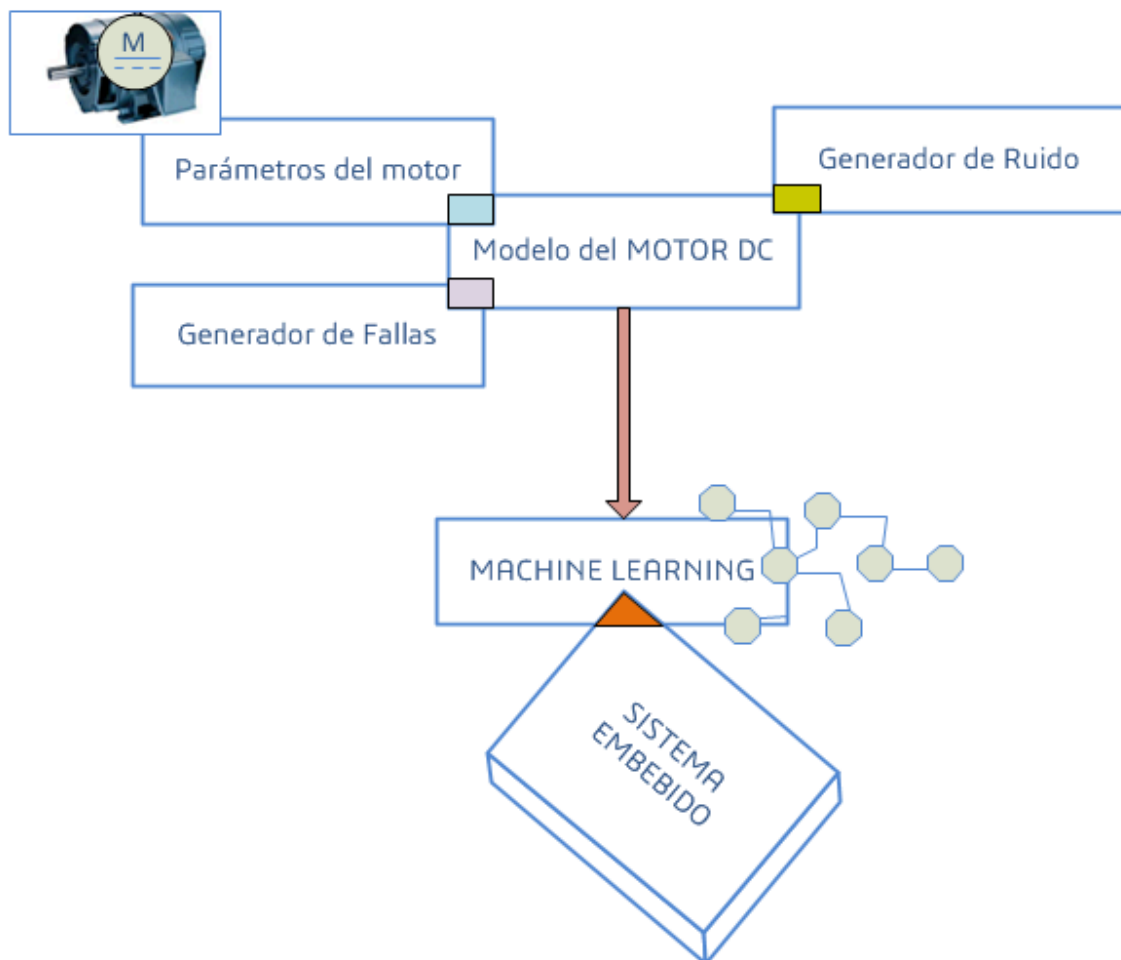
Se pretende mostrar las mejoras en el mantenimiento predictivo al basarse en machine learning para los sistemas de diagnósticos de fallas de motores de corriente directa. Figura 2.

Por lo cual utiliza el modelo del motor DC, generadores de falla y ruido para simular un entorno típico de operación que permite configurar las características de los escenarios de prueba y finalmente realizar el despliegue del modelo obtenido del sistema de diagnóstico en un sistema embebido.

### a. Estructura general

**Figura 2**

*Proceso de Sistema de Diagnóstico*



## **b. Explicación del aporte**

Las actividades que generan interactividad, representan el uso de un switch manual y el valor de una constante dentro del programa de Simulink que permite seleccionar el tipo de falla para obtener el dato de prueba de referencia del modelo de motor DC y falla para propósitos de validación del proyecto.

Se utilizaron los recursos de herramientas como MATLAB y QUARTUS en las siguientes actividades:

### **Diseño del programa en MATLAB**

El programa se encuentra dividido en un script con secciones de código para las tareas de: parámetros (figura 4) y Machine learning (Figura 5) y en un modelo en Simulink (Figura 3) para el modelo del motor DC y el generador de fallas y ruido.

#### Elementos utilizados en Simulink

Constantes, Sumador, Ganancia, Integrador, Switch, Multiplicador, Builder Signal, Scope, To Workspace, Extract Bit, Band Limit White Noise.

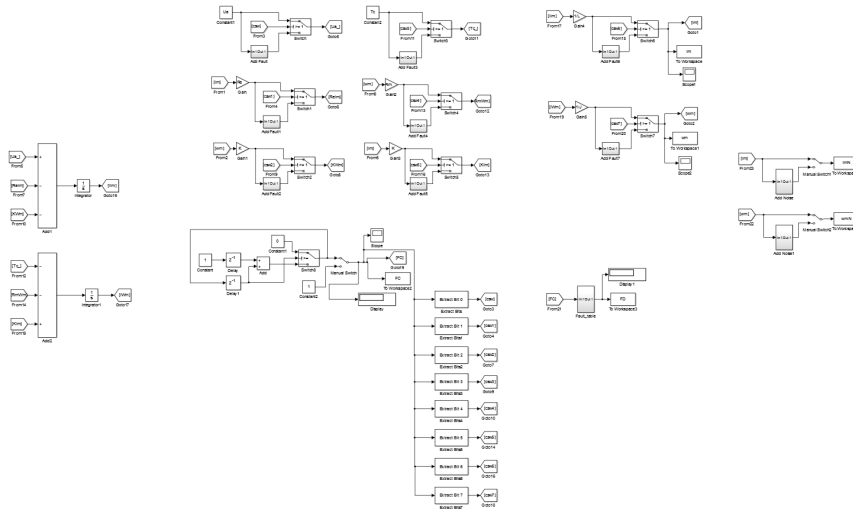
#### Funciones utilizadas en Matlab

Classdef, nntools, Neural Net Fitting App (MATLAB, 2024), Neural Net Pattern Recognition (MATLAB, 2024).

### **Programa realizado en MATLAB/Simulink**

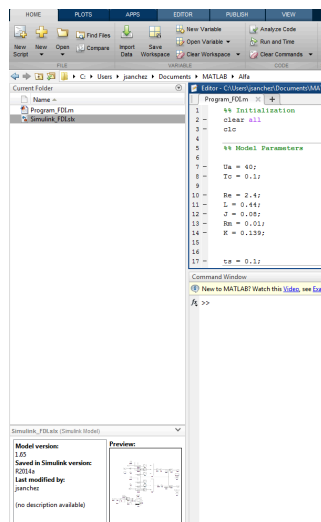
**Figura 3**

*Programa Simulink*



**Figura 4**

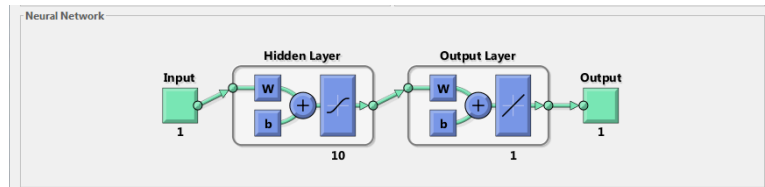
*Programa MATLAB*



## Red neuronal realizada en MATLAB

**Figura 5**

*Arquitectura de Red Neuronal*



Las actividades que se realizan para la evaluación son:

-Comparar el resultado del error para modelos machine learning establecidos del sistema de detección de fallas.

-Comparar el resultado obtenido del modelo del motor dc y del modelo de red neuronal para un dato particular de prueba del sistema de diagnóstico de fallas.

### **c. Estrategias y/o técnicas**

En el proceso de diseño se emplearon las siguientes estrategias y técnicas:

#### **Creación del modelo del motor DC**

A partir de las ecuaciones de corriente y velocidad del motor, se realiza el diagrama de componentes (Figura 6) para el modelo de comportamiento ideal del motor DC. Tabla 1.

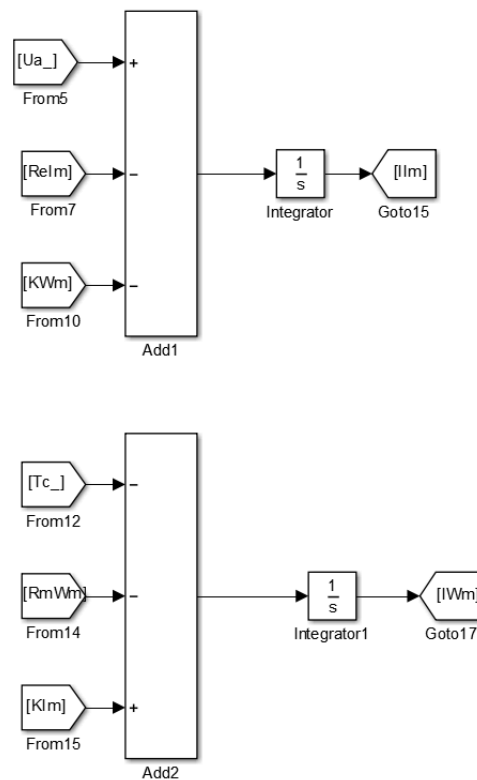
**Tabla 1**

*Parámetros y Entradas del Motor DC*

Parámetro/Entrada	Simbología	Unidades (SI)
Voltaje	Ua	V
Torque externo	Tc	Nm
Resistencia eléctrica	Re	Ohm
Constante del motor	K	Nm/A
Inductancia eléctrica	L	H
Resistencia mecánica	Rm	Ns/m
Inercia Rotacional	J	Kgm2

**Figura 6**

*Modelo Motor DC*

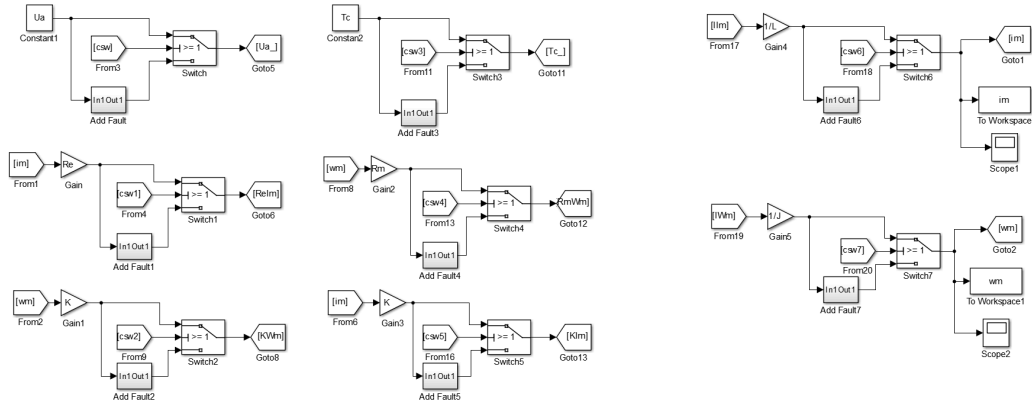


**Creación de Generador de Fallas**

Se establece un patrón de fallas llamado –Offset- para la magnitud de los valores de los parámetros y las entradas del motor DC. Figura 7.

Figura 7

Generador de Fallas

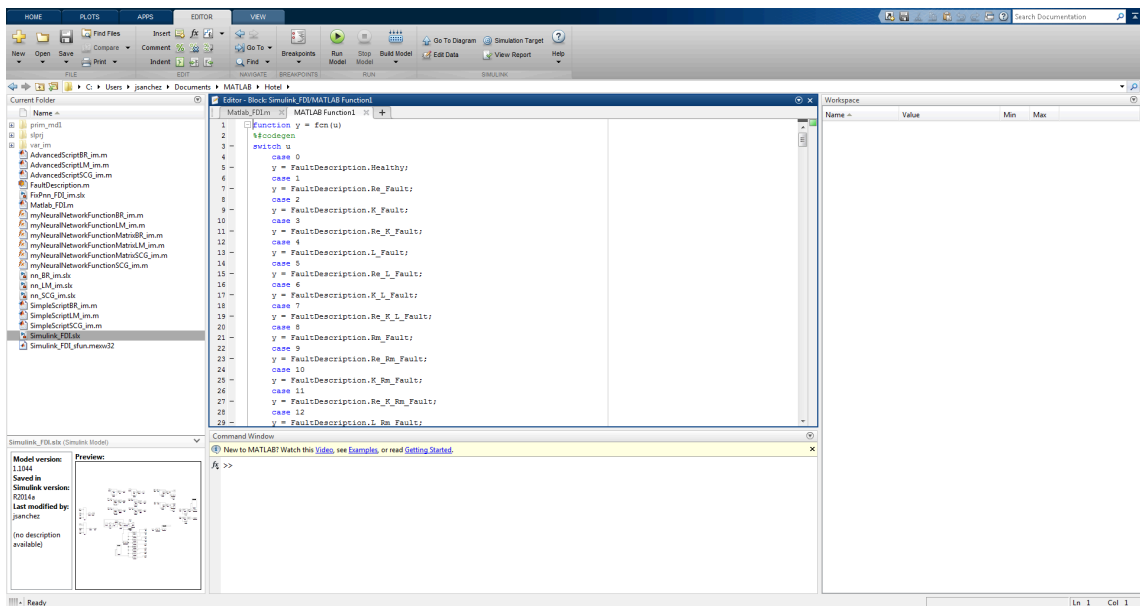


### Creación de identificadores de Fallas

Establecimiento de una tabla de fallas para la combinación de las fallas del motor. Figura 8.

Figura 8

Función de Identificador de Fallas



## Generación de datos para red neuronal

Se identifican los datos para entrada y objetivo, necesarios para la red neuronal Figura 9.

### Figura 9

*Predictores y Resultados de la Red*

---

```
%% Input and target data for neural network of motor current
x_1 = Inp_i;
t_1 = Tar;
```

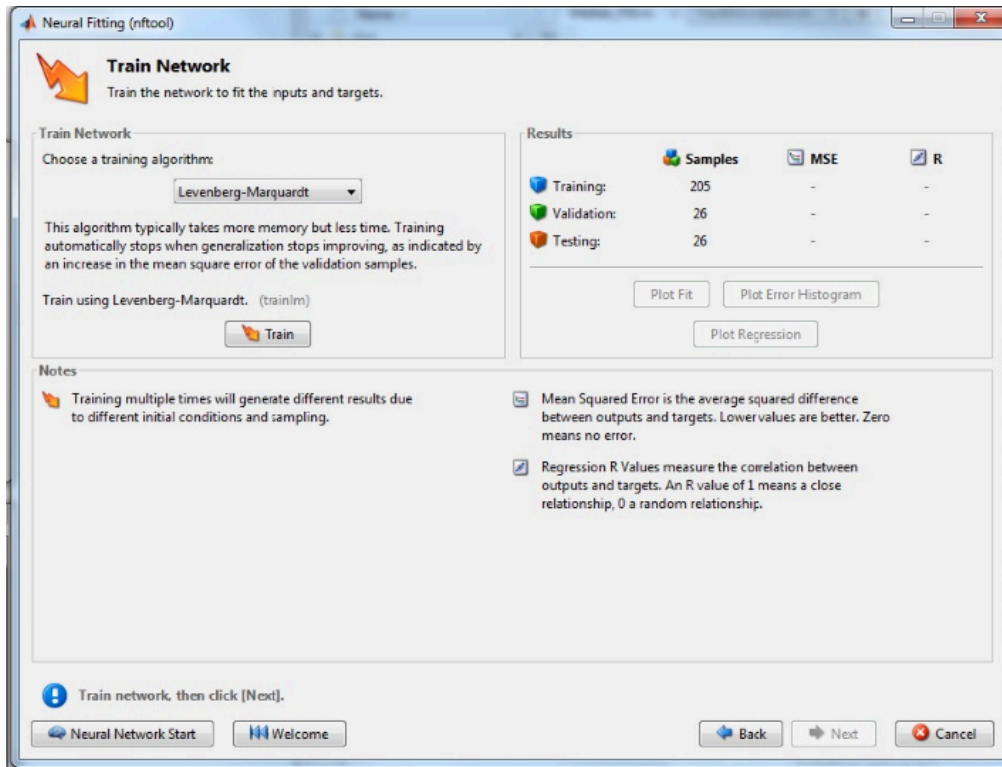
---

## Ajuste de la Red

Distribución de datos para entrenamiento, validación y test. Configuración del número de neuronas y algoritmo de entrenamiento de la red neuronal Figura 10.

**Figura 10**

*Configuración del Entrenamiento*



**Prueba del Sistema de Diagnóstico**

A partir de los datos de prueba se realiza la comprobación del sistema de diagnóstico utilizando gráficas y resultados numéricos que permiten considerar la precisión del sistema.

Se utiliza la herramienta Matlab/Simulink para el diseño de Modelos y análisis de datos.

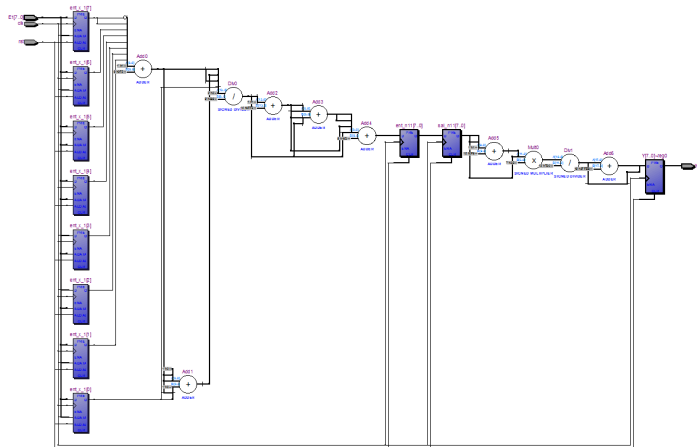
**Creación de Red Neuronal en Quartus a partir del modelo producido por MATLAB**

Se utiliza la herramienta Quartus para el diseño del Modelo en VHDL Figura 11.



**Figura 11**

*Red Neuronal en Quartus*



**2.3 Validación de la propuesta**

Una vez elegido el modelo de la red en MATLAB, Tabla 2, se procedió a realizar la comprobación Tabla 3, con la ayuda del modelo FDI de Simulink y el modelo de la red en Quartus, así se realizó tres escenarios de falla (Figura 13, Figura 14, figura 15, Figura 16, figura 17 y Figura 18) que permiten utilizar valores de prueba Figura 12.

**Tabla 2**

*Datos de la red*

Modelo	Función de entrada	Función de salida	Número de neuronas de entrada	Número de neuronas de salida
Levenberg-Manquardt	tansig	purelin	10	1

**Tabla 3**

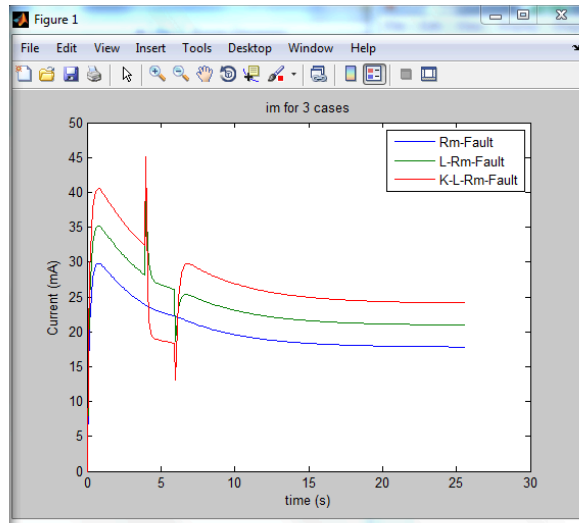
*Comparación entre modelos de red*

Número de escenario	Salida en FDI simulink	Salida en red MATLAB	Salida en red Quartus
1	8	8,72	6
2	12	12.35	11
3	15	14.97	13

Las diferencias corresponden al manejo de valores de los datos en algunos enteros o decimales y también si este es redondeado o si es punto fijo o punto flotante.

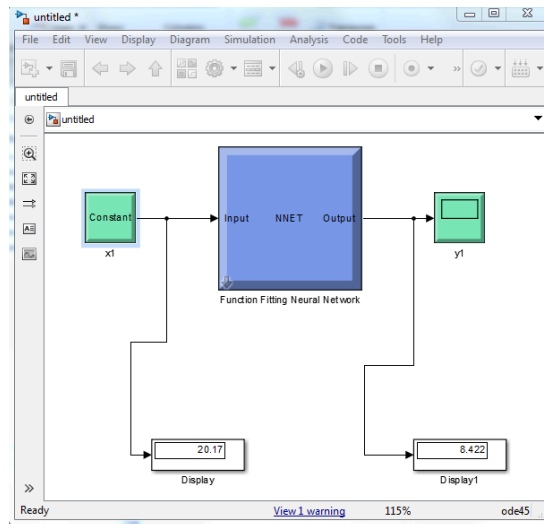
**Figura 12**

*Vectores de Prueba - Matlab*



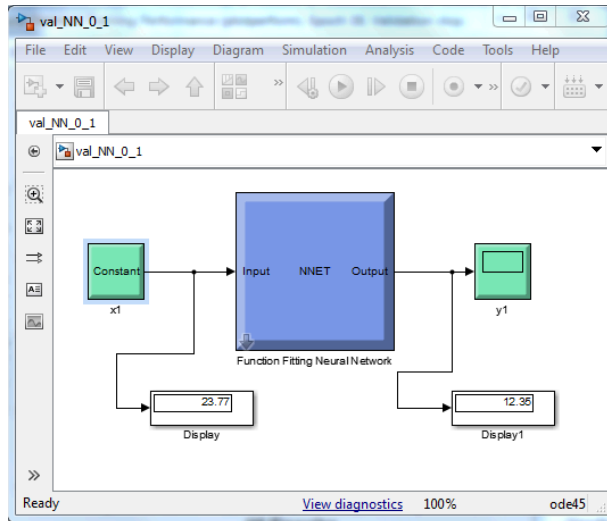
**Figura 13**

*Escenario de prueba 1 - Simulink*



**Figura 14**

*Escenario de prueba 2 - Simulink*



**Figura 15**

*Escenario de prueba 3 - Simulink*

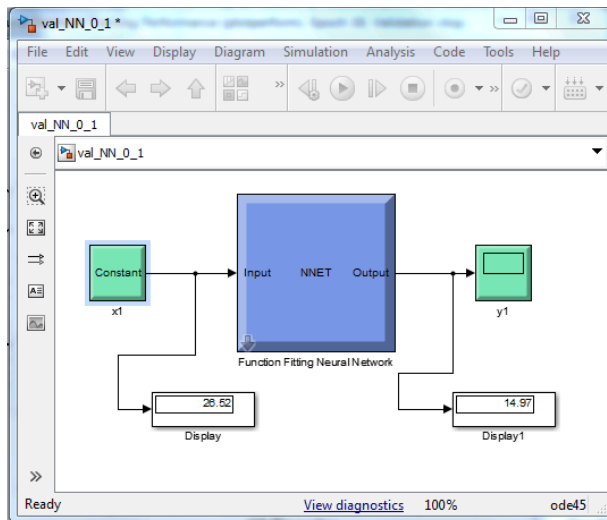


Figura 16

Vector de prueba y escenario de prueba 1 - Quartus

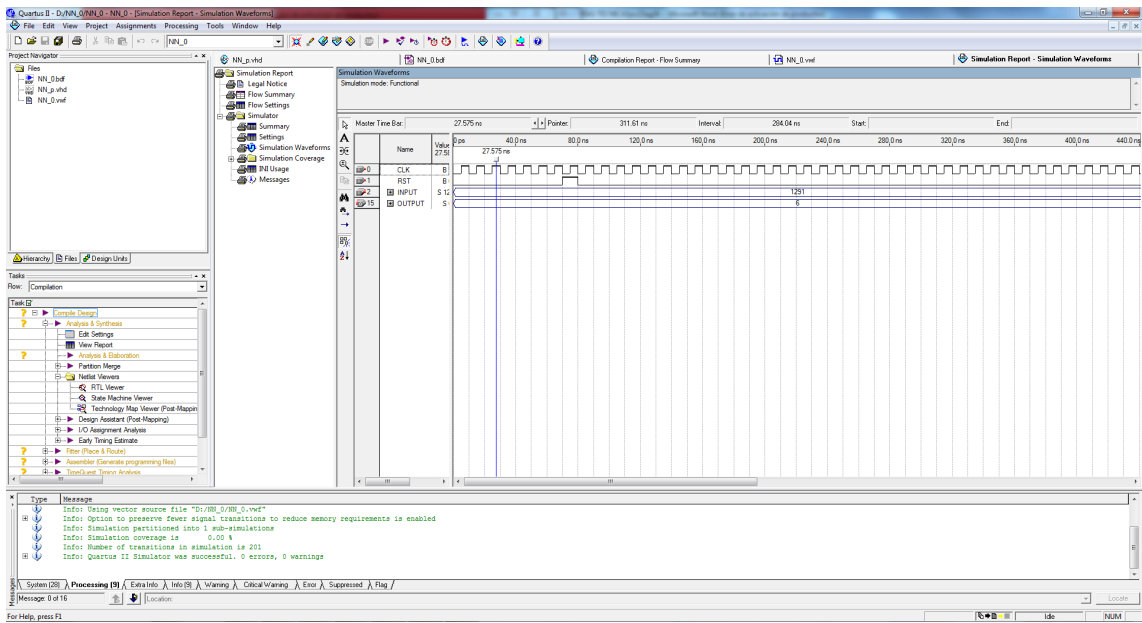


Figura 17

Vector y escenario de prueba 2 - Quartus

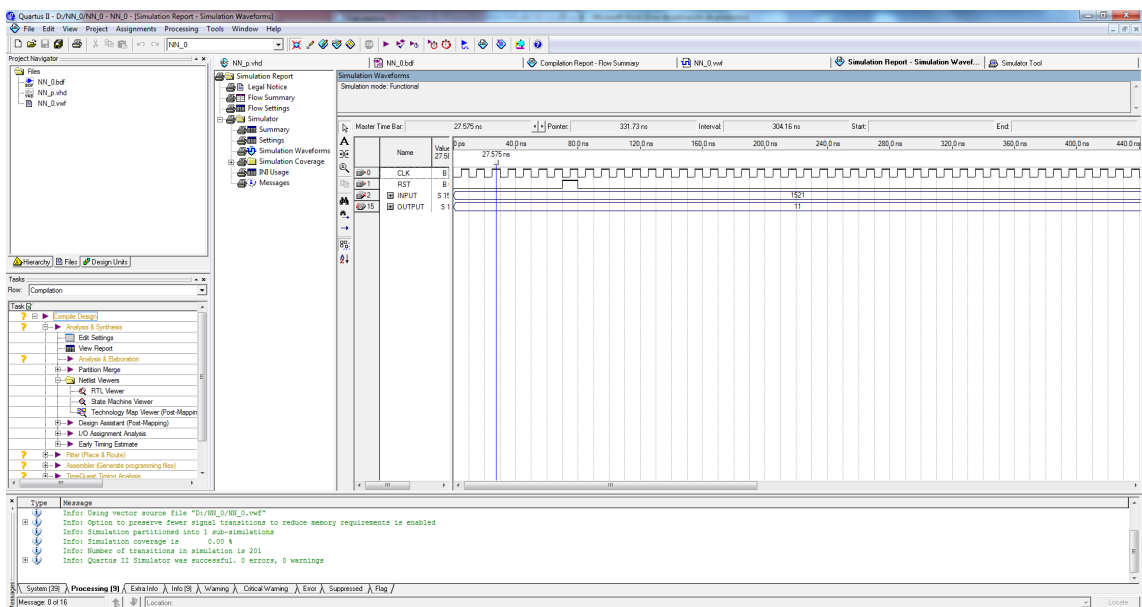
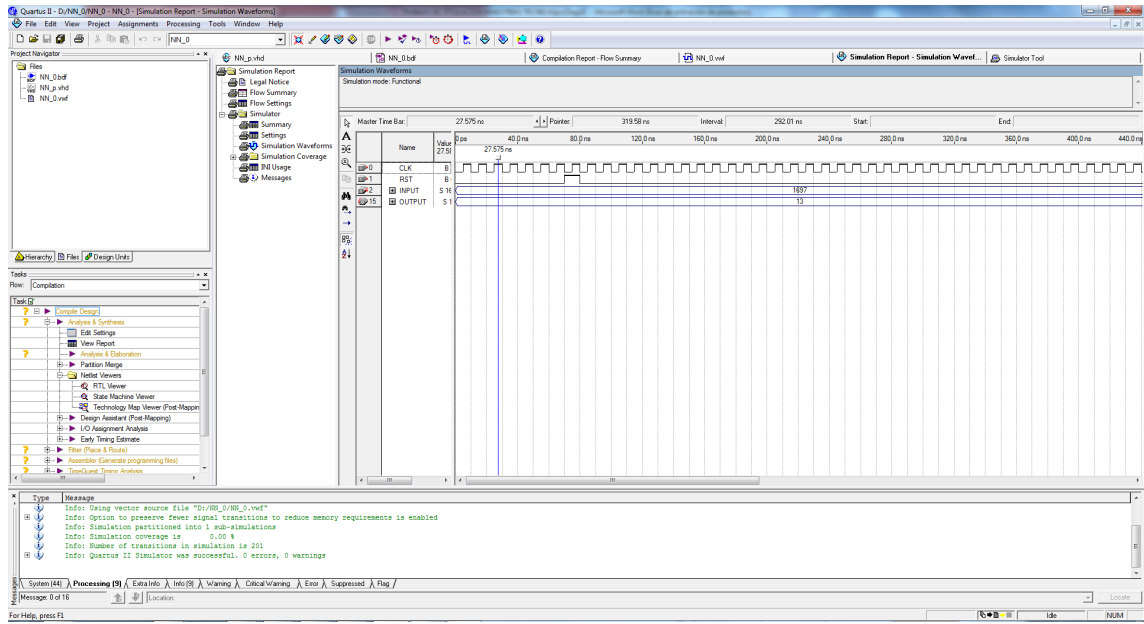


Figura 18

Vector y escenario de prueba 3 - Quartus



## 2.4 Matriz de articulación de la propuesta

Resumen de lo relacionado al sistema Tabla 4:

**Tabla 4**

*Matriz de articulación*

Elementos de lo propuesto	Breve descripción de los resultados de cada parte	Justificación de teorías relacionadas a la propuesta	Fórmulas ó técnicas utilizadas
<b>1</b> Creación de modelo del motor Creación de Generador de Fallas Creación de identificador de fallas. Generación de datos para red en Matlab. Ajuste de la red. Prueba del Sistema de Detección Creación de red en Quartus	Presentación en gráficos Revisión de valores Elección basada en menor error	Matemáticas Aplicadas Machine Learning	Ecuaciones diferenciales Leyes: Kirchoff y dinámica Aprendizaje supervisado Regresión Lineal
<b>2</b> Estructura y división por bloques y funciones	Aplicación en Matlab Aplicación en Quartus	Programación de FPGA Aplicaciones de diseño de circuitos electrónicos (Proteus, Isis, Ares) Matlab, Quartus.	Bloques de Funciones matemáticas Generación de señales Conversión de datos Apps de ML
<b>3</b> Datos requeridos por los escenarios de prueba	Resultados de cada bloque y función	Análisis de Señal	Vector de Prueba Herramienta de Simulación Funcional

## **2.5 Análisis de resultados. Presentación y discusión.**

### **a. Contextualización de los fundamentos teóricos.-**

#### *Sistema de Detección de Fallas*

Es un sistema encargado de detectar e identificar las fallas utilizando mediciones al objeto que analiza. Se lo conoce como Fault Detection and Identification (FDI) y es muy utilizado para los procesos o sistemas. Se dividen en dos métodos que son llamados deductivo e inductivo.

El método deductivo se basa en principios generales y es más preciso que el inductivo, pero el inductivo es más realizable. El inductivo se basa en patrones de los datos del proceso, siendo las redes neuronales parte de este grupo.

#### *Uso de Machine Learning (ML)*

Como la técnica de ML emplea el entrenamiento para obtener la relación de la salida y la entrada, al usar datos de entrenamiento en condición normal, la condición de falla se puede detectar al notar la desviación del valor medido respecto al estimado de una variable del sistema.

### **b. Diseño del Sistema de Detección de Fallas**

El bloque del sistema posee una entrada y una salida. Donde la entrada es la corriente y/o la velocidad del motor y la salida es el código de falla. Con la ayuda del generador de fallas se producen los cambios en el comportamiento del motor que permiten realizar el registro de las señales correspondiente a cada falla y combinar fallas entre sí, por lo cual existe una asociación del síntoma de falla con el código. Para este estudio, el sistema de detección se diseñó a nivel de fallas internas (parámetros del motor) y externas (actuador y carga).

### **c. Análisis de los parámetros estadísticos y de clasificación.-**

Para clasificar los datos se utilizan las características más relevantes de la señal para ser los predictores que ayuden en el desempeño del modelo. Se realizó el cálculo de las características típicas y aquellas relacionadas al procesamiento de señal para finalmente seleccionar una sola característica (media) que permite comprobar el modelo de detección y optimizar el uso de memoria de cálculo. Tabla 5.

**Tabla 5***Comparación de características*

Media	Standard Deviation	Skewness	Kurtosis	Fault Description
20.1742	3.4520	0.4313	7.8353	'Rm-Fault'
23.7658	4.2290	0.6155	7.5353	'L-Rm-Fault'
26.5199	5.2363	0.5499	6.4692	'K-L-Rm-Fault'

El modelo elegido para la clasificación es la red neuronal y se realizaron pruebas preliminares con el fin de obtener el mejor modelo, variando el algoritmo de entrenamiento y el número de neuronas, además se realizó la variación del porcentaje de división de datos de entrenamiento, validación y pruebas, finalmente se fijó un porcentaje de 80, 10,10 para a partir de esto, realizar la búsqueda del modelo adecuado.

Lo cual se aprecia en la Tabla 6:

**Tabla 6***Comparación de modelos Machine Learning*

Modelo	Algoritmo de entrenamiento	Número de Neuronas en Capa -Escondida-	Epoch	Error
1	Levenberg-Marquardt	10	29	7.5911
2	Bayesian Regularization	10	45	6.701
3	Scale Conjugate Gradient	10	45	10.7587

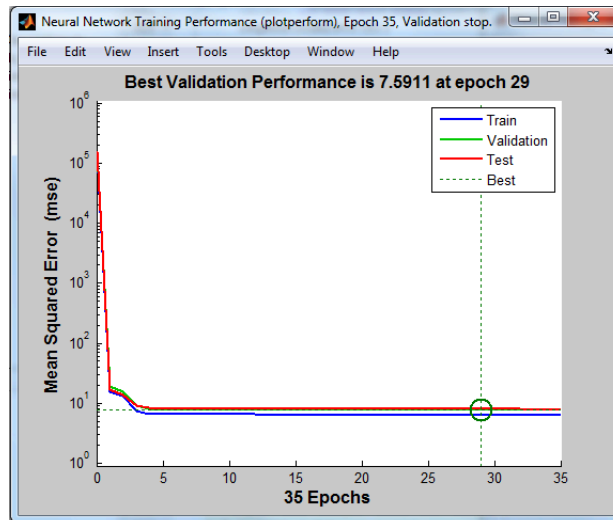
Y se puede observar en las curvas de desempeño para la variable de corriente:



-Levenberg-Marquardt Figura 19

**Figura 19**

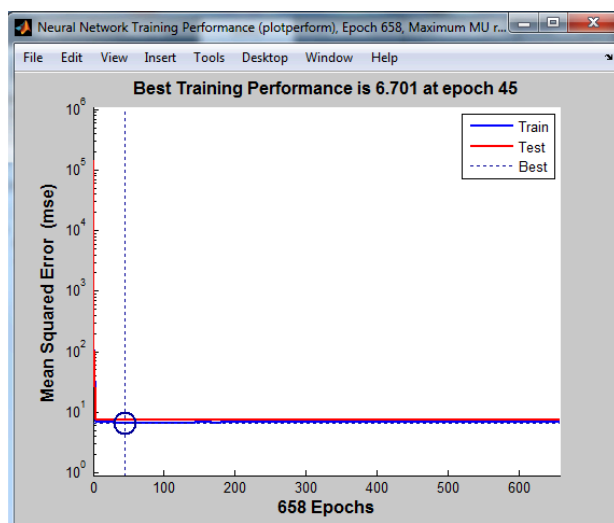
*Performance Levenberg-Manquardt*



-Bayesian Regularization Figura 20

**Figura 20**

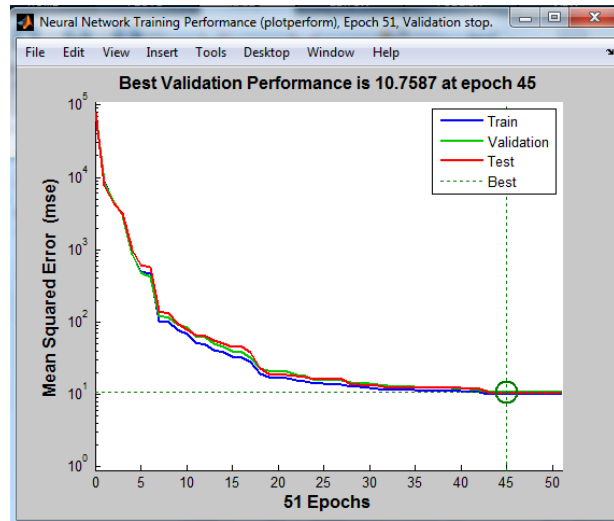
*Performance Bayesian Regularization*



-Scaled Conjugate Gradient Figura 21

**Figura 21**

*Performance Scaled Conjugate Gradient*



Se puede observar el desempeño Figura 22 y el histograma de error Figura 23 del modelo elegido en el segundo entrenamiento:

Figura 22

Performance Levenberg-Manquardt (elegido)

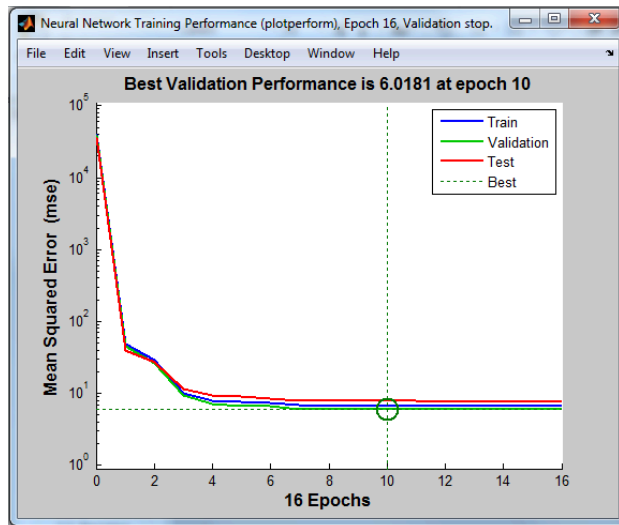
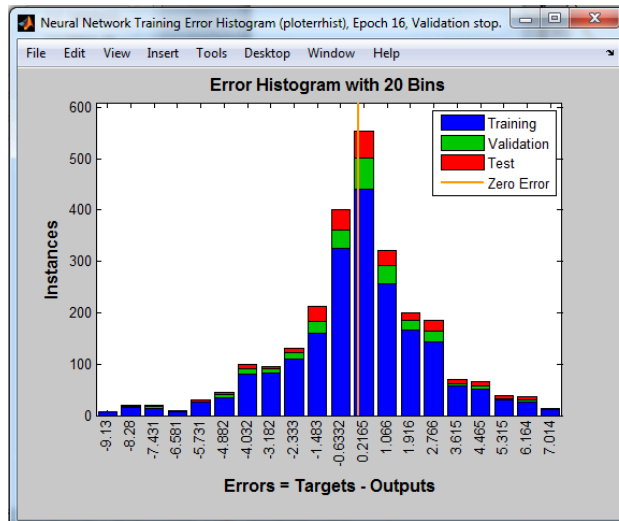


Figura 23

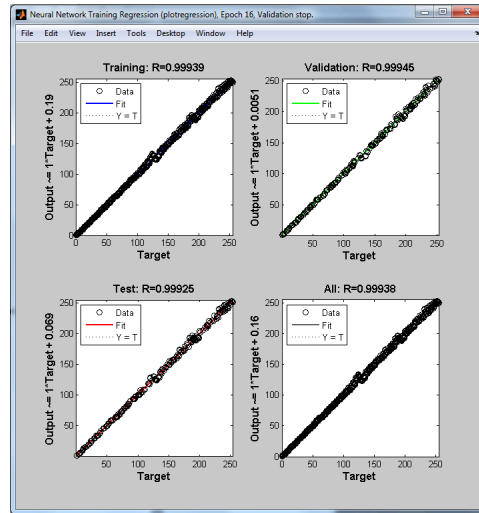
Error Histogram



Donde existen 20 contenedores y los datos se acercan alrededor del error de 0.2 para la comparación entre el valor objetivo y previsto.

El gráfico de regresión Figura 24:

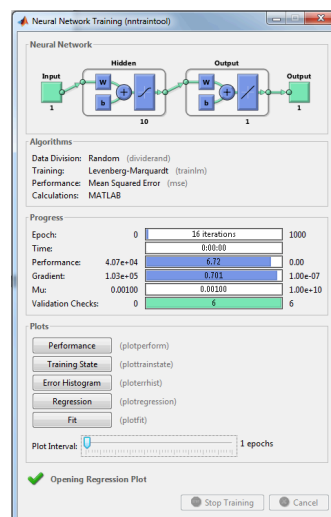
**Figura 24**  
Regression



Vemos que la regresión es cercana a 1 y es prácticamente lineal, consiguiendo que el modelo encaje bien con los datos.

Los resultados del entrenamiento Figura 25:

**Figura 25**  
Resultados del entrenamiento

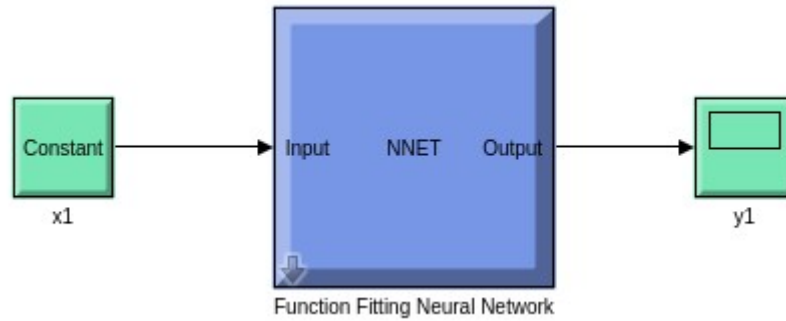


**d. Validación del modelo.-**

Y el modelo de la red neuronal obtenida Figura 26

**Figura 26**

*Red Neuronal para el Sistema*



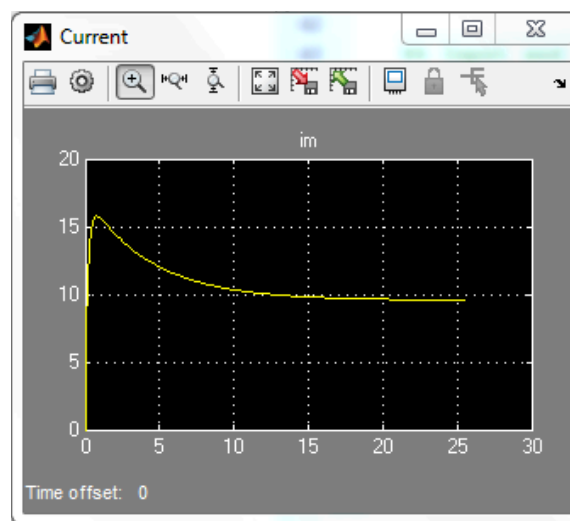
Se realizan tres pruebas para observar estado del motor con las condiciones de: Rm\_Fault, L\_Rm\_Fault, K-Fault

Así también se observa las señales del estado del motor en condición normal y en condición de falla:

-Modo: -Healthy- Figura 27, Figura 28

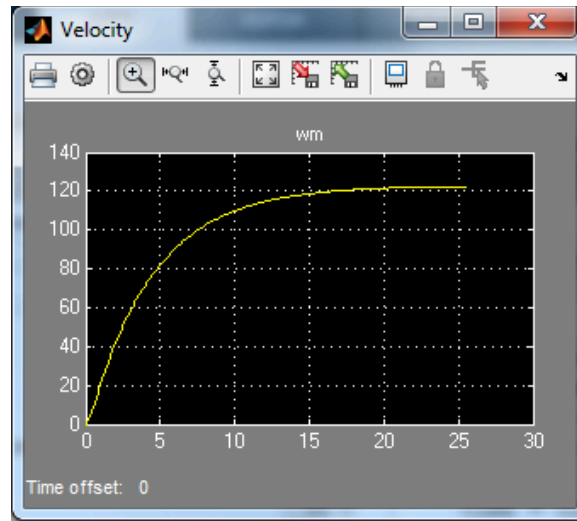
**Figura 27**

*Gráfico im -Healthy-*



**Figura 28**

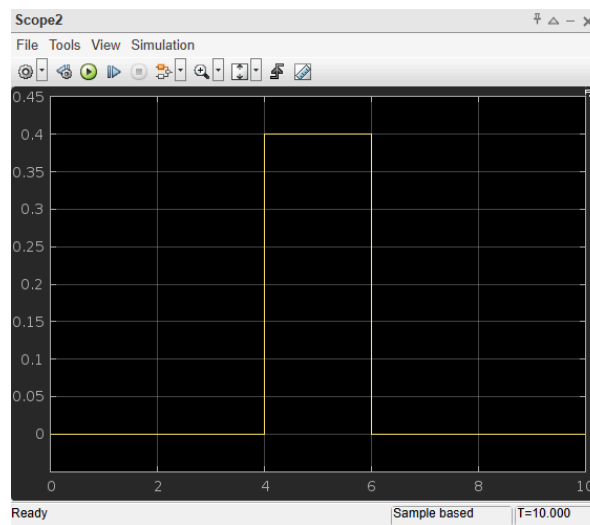
*Gráfico  $w_m$  -Healthy-*



-Falla del offset de 40% en la Resistencia mecánica del motor ( $R_m$ ) – $R_m$  Fault- Figura 29

**Figura 29**

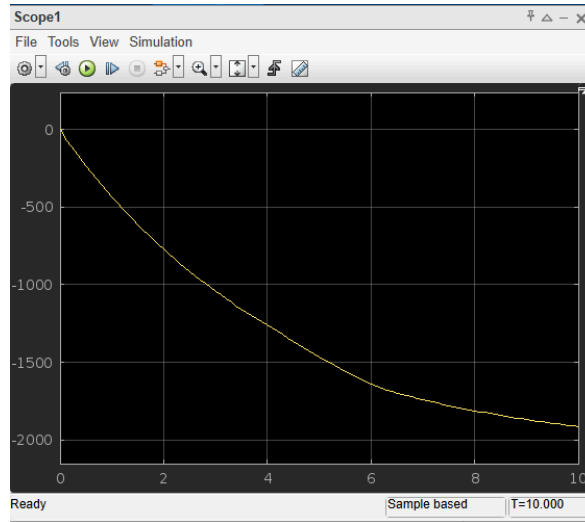
*Gráfico de Señal de Falla de Offset*



Y su efecto en la corriente del motor Figura 30

**Figura 30**

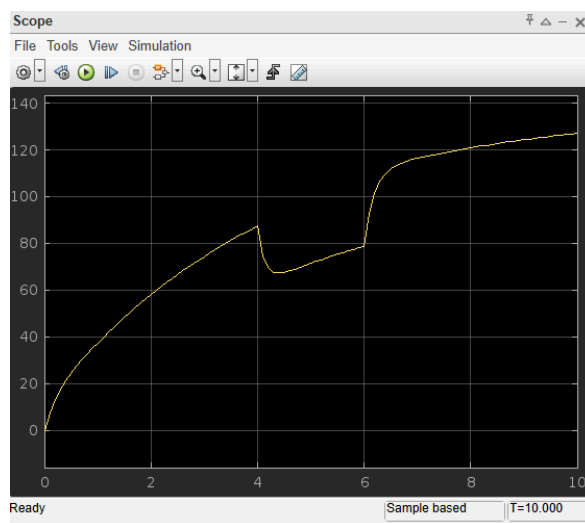
*Gráfico de  $i_m$  -Rm-Fault-*



Y en la velocidad del motor Figura 31:

**Figura 31**

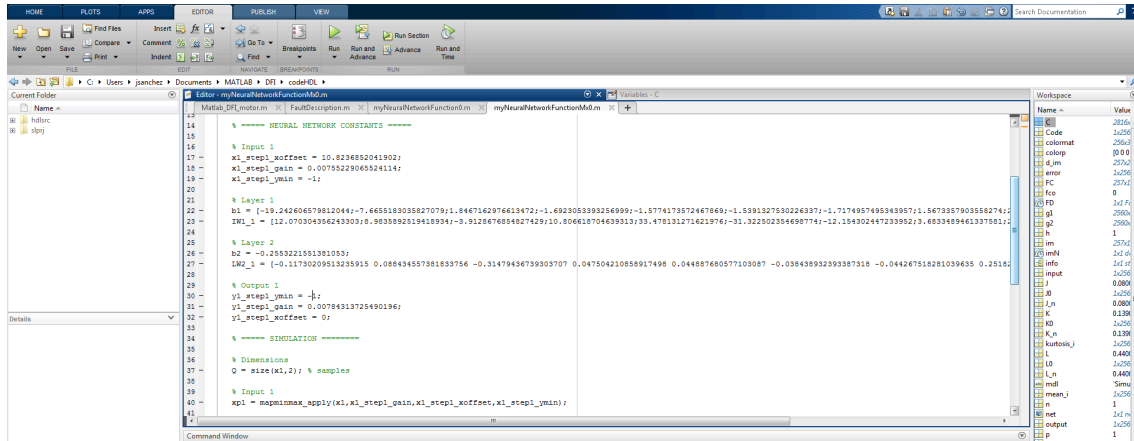
*Gráfico de  $w_m$  -Rm-Fault-*



Y a partir de los valores de los parámetros de la red neuronal Figura 32, obtenidos de su función

Figura 32

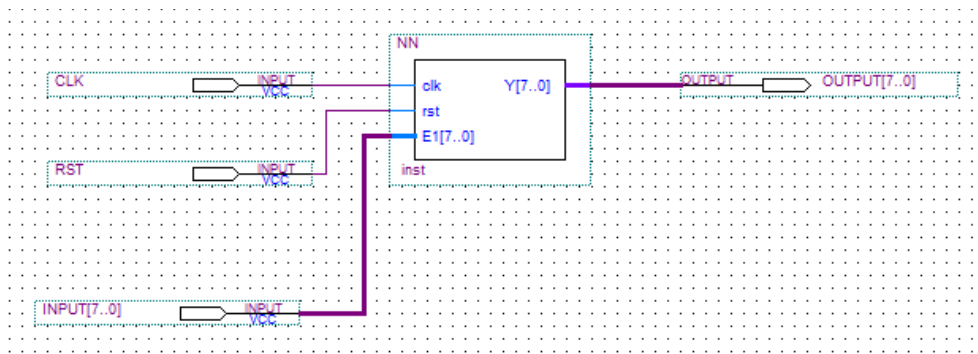
Parámetros de la red neuronal



Se pudo generar el modelo para el despliegue en el sistema embebido Figura 33:

Figura 33

Bloque de Red neuronal -Quartus



Y se utilizó los mismos vectores de prueba que se empleó para la validación del modelo en matlab/simulink.



## CONCLUSIONES

En esta investigación se pudo comprobar cómo el uso de la tecnología adecuada puede facilitar las tareas de procesamiento y análisis de información. Debido a esto el ANN es muy utilizado para diversos campos.

El motor con ANN se desarrolló para la detección de fallas internas y externas, que pueden surgir en forma aislada o en grupo desde dos hasta ocho fallas a la vez.

La Detección de Fallas en el motor siguiendo la estructura teórica FDI y utilizando la técnica de ML permitió adquirir conocimiento sobre su aplicación en procesos industriales.

Se consigue realizar con éxito el diseño del sistema de detección de fallas del motor, empleando machine learning, Además se diseña para que pueda ser utilizado en un dispositivo embebido.

Se pudo realizar la comparación de dos modelos de machine learning y se pudo determinar que a pesar que el algoritmo de “Bayesian Regularization” obtuvo un menor error para detección de fallas, se elige el modelo “Levenberg-Manquardt” por ser más rápido y por tener también un menor error en la predicción.

Las fallas del motor se predicen bien con la ayuda del modelo de machine learning lo cual se comprueba con la métrica de precisión del modelo para las variables de corriente y velocidad, lo cual se observa con los resultados de la salida obtenida del modelo ANN a ser utilizado.

## RECOMENDACIONES

El trabajo se fundamenta en el conocimiento previo de los valores de los parámetros del motor, que pueden ser obtenidos de forma experimental con la ayuda de equipos de adquisición de datos y análisis matemático como lo han realizado varios autores en diversas investigaciones lo cual no es materia de este trabajo, pero para futuras investigaciones si se requiere realizar un diseño completo del motor debería abarcarlo. , El diseño se despliega para un sistema embebido, permitiendo que con nuevas investigaciones pueda ser integrado en un sistema de monitoreo y control del motor en tiempo real, donde las limitaciones que se tendrían corresponden al tamaño de la memoria y el tamaño del código del dispositivo embebido, por lo cual se hace necesario optimizar los recursos.

Existe una amplia combinación de variaciones para configurar modelos de machine learning, y los resultados de precisión de la salida depende de los datos, la versión del software, los recursos de la computadora que obtiene el modelo, el número de veces del entrenamiento del modelo, por lo que deberá revisarse estas consideraciones cuando se requiera encontrar el modelo adecuado para una aplicación particular.

El estudio realizado corresponde a simulaciones de fallas de los parámetros eléctricos y mecánicos pero no considera otros fallos como vibración o temperatura, lo cual a futuro sería complementario para tener un sistema robusto de diagnóstico del motor.

## BIBLIOGRAFÍA

- Linkedin. (2024). *Uso de machine learning para el mantenimiento predictivo* [Página web].  
<https://es.linkedin.com/pulse/uso-de-machine-learning-para-el-mantenimiento-la-con-elies-fmqtf>
- AMFEC. (2015). *Técnicas Predictivas para Detectar Fallos en Motores de Corriente Continua* [Página web].  
<https://www.ingenieriamantenimiento.org/detectar-fallos-motores-de-corriente-continua/>
- International Couplings. (s/f). *Motores de corriente continua: usos actuales* [Página web].  
<https://internationalcouplings.es/blog/motores-de-corriente-continua-usos-actuales>
- Sardashti, A., Nazari, J. (2023). A learning-based approach to fault detection and fault-tolerant control of permanent magnet DC motors. *Journal of Engineering and Applied Science*, 70-109 (2023).  
<https://doi.org/10.1186/s44147-023-00279-5>
- Bhatia, A., Vishwakarma, V. (2023). Machine Learning for Fault Detection in DC Motors and the role of Mounting Configurations. *2023 IEEE International Conference on Computer Vision and Machine Intelligence (CVMI)*, 1-5. <https://doi.org/10.1109/CVMI59935.2023.10465171>
- CORDIS. (2021). *Llevar el mantenimiento predictivo a la planta de producción* [Página web].  
<https://cordis.europa.eu/article/id/435256-bringing-predictive-maintenance-to-the-factory-floor/es>
- Cabal-Yépez, E. et al. (2014). Real-time condition monitoring on VSD-fed induction motors through statistical analysis and synchronous speed observation. *Research on Online Library Wiley*, 25(8), 1657-1672,  
<https://onlinelibrary.wiley.com/doi/full/10.1002/etep.1938>
- Restrepo et al. (2017). Metodología de detección de fallas de un motor síncrono. *Investigación en Revista Colombiana de tecnologías de avanzada*, 2(26), [https://revistas.unipamplona.edu.co/ojs\\_vic](https://revistas.unipamplona.edu.co/ojs_vic)
- Jong-Hyun et al. (2019). Fault diagnosis of Induction Motor using Convolution Neural Network. *Research on MDPI Open Access Journal*, 9(15), 2950,  
[https://www.mdpi.com/2076-3417/9/15/2950?type=check\\_update&version=1](https://www.mdpi.com/2076-3417/9/15/2950?type=check_update&version=1)
- Vicente et al. (2020). Stator Winding Fault Detection Using External Search coil and Artificial Neural Network. Dissertation, MATEC Web of Conferences. 01054.
- Castro, S. (2022). *Técnicas de Machine Learning aplicadas en modelos de mantenimiento predictivo en empresas industriales*. [Tesis de Grado, Universidad Industrial de Santander]. Repositorio digital de tesis.

<https://noesis.uis.edu.co/server/api/core/bitstreams/f2b27c2e-c888-4b5d-a64a-79ee7697a87a/content>

Morales, V. (2023). *Machine Learning: Teoría y Práctica* [Página web].

[https://bookdown.org/victor\\_morales/TecnicasML/introducci%C3%B3n.html](https://bookdown.org/victor_morales/TecnicasML/introducci%C3%B3n.html)

Freire, J. (2023, 20 de Junio). 'La inteligencia artificial todavía es de ciencia ficción', dice experto en ciberseguridad, y agrega que el centro de atención está en el 'machine learning'. *El Universo*.

Descargado el 02 de Agosto de 2024 de

<https://www.eluniverso.com/larevista/tecnologia/la-inteligencia-artificial-todavia-es-de-ciencia-ficcion-dice-experto-en-ciberseguridad-y-agrega-que-el-centro-de-atencion-esta-en-el-machine-learning-nota/>

Cárdenas, M. (2022). Sistema de realidad aumentada para mantenimiento preventivo del sistema de generación en la central hidroeléctrica Sopladora. Recuperado de

<https://repositorio.uisrael.edu.ec/xmlui/handle/47000/3411>

Castillo, B. (2022). Sistema de detección predictiva de fallas de un motor de baja tensión mediante protocolo MQTT y aplicación IoT. Recuperado de <https://repositorio.uisrael.edu.ec/xmlui/handle/47000/3317>

Arequipa, E. (2022). Sistema de realidad aumentada para mantenimiento correctivo industrial. Recuperado de <https://repositorio.uisrael.edu.ec/xmlui/handle/47000/3311>

IBM. (s/f). *¿Qué son las redes neuronales?* [Página web]

<https://www.ibm.com/es-es/topics/neural-networks>

BBVA. (2024). Machine learning: ¿qué es y cómo funciona el maestro de reconocer patrones?

<https://www.bbva.com/es/innovacion/machine-learning-que-es-y-como-funciona/>

SAP. (s/f). *¿Qué es machine learning?* [Página Web]

<https://www.sap.com/latinamerica/products/artificial-intelligence/what-is-machine-learning.html#:~:text=El%20machine%20learning%20se%20compone,%2C%20semisupervisado%2C%20o%20de%20refuerzo>

Aprendemos Tecnología. (2010). Motores eléctricos Parte 2

<https://aprendemostecnologia.org/wp-content/uploads/2010/02/motores-electricos-parte-ii1.pdf>

Arana, A. (2021). Aplicación de la Inteligencia Artificial en la detección de fallas en los motores eléctricos de corriente continua de imán permanente.

[https://repositorio.uncp.edu.pe/bitstream/handle/20.500.12894/7313/T010\\_48040651\\_T.pdf;jsessionid=4B23AB2CFD256CD1F8C8F806AECDE003?sequence=1](https://repositorio.uncp.edu.pe/bitstream/handle/20.500.12894/7313/T010_48040651_T.pdf;jsessionid=4B23AB2CFD256CD1F8C8F806AECDE003?sequence=1)

Brito, W. (s/f). *Informe 6* [Página web].

<https://www.coursehero.com/file/66647207/LABCONV-GR7-BRITO-WILSON-INFORME-6pdf/>

Scalab. (s/f). Redes. [Diapositivas de Power Point]. Repositorio de SCALAB Universidad Carlos III de Madrid.

<http://scalab.uc3m.es/~docweb/it-sup/transparencias/redes.pdf>

Gracia, A., Tainta, S. (2017). *Diseño y construcción de un robot auto-balanceado mediante Arduino*. [Tesis de Grado, Universidad Pública de Navarra]. Repositorio digital de tesis.

<https://academica-e.unavarra.es/server/api/core/bitstreams/47bdf091-493a-46ef-a3dc-f681f8eabdb9/content>

MATLAB. (2024). *Centro de ayuda Mathworks* [Página web].

<https://la.mathworks.com/help/deeplearning/ref/nnstart.html?action=changeCountry&nocookie=true>