

Research Article

Development of a Method for Diagnosing Faults in Hydraulic Systems Using Artificial Neural Networks with Deep Learning

Desarrollo de un método de diagnóstico de fallas en sistemas hidráulicos utilizando redes neuronales artificiales con aprendizaje profundo

Hugo Mera*, Félix García, Edisson Calderón

Investigador independiente, Escuela Superior Politécnica de Chimborazo (ESPOCH), 060155, Riobamba, Ecuador.

V CONGRESO
INTERNACIONAL DE
INGENIERÍA DE MECÁNICA

Corresponding Author: Hugo
Mera; email:
merahugo1d@gmail.com

Published: 24 July 2024

Production and Hosting by
Knowledge E

© Hugo Mera et al. This article is distributed under the terms of the [Creative Commons Attribution License](#), which permits unrestricted use and redistribution provided that the original author and source are credited.

Abstract

The application of artificial intelligence is a recent improvement in the industry, allowing preventive maintenance to be applied as a reliability method for detecting failures in hydraulic systems. This is achieved by using artificial neural networks (ANN) as classifiers to make automatic binary and categorical decisions. Since these systems have multiple conditions and sub-conditions that can be complex for normal analysis, the UCI repository database is relied upon to construct an intelligent algorithm of artificial neural networks with deep learning. This has proven to be a highly effective way of predicting failures, with an overall accuracy rate of 97% when applying the intelligent model to the system. As a result, it can be concluded that deep learning is much more efficient than classical machine learning.

Keywords: artificial intelligence, predictive maintenance, artificial neural networks, deep learning.

Resumen

La aplicación de la inteligencia artificial es la nueva mejora en la industria, permitiendo que el mantenimiento preventivo se aplique como método de confiabilidad para la detección de fallos en sistemas hidráulicos aplicando Redes neuronales artificiales (ANN), utilizándoles como clasificadores para obtener una toma de decisiones automáticas de manera binaria y categórica, ya que dichos sistemas poseen varias condiciones y subcondiciones que se vuelven complejas para un análisis normal, apoyándose en la base de datos del repositorio de la UCI, siendo analizados para la construcción de un algoritmo inteligente de redes neuronales artificiales con Deep Learning (aprendizaje profundo), demostrando así un alto desenvolvimiento en la predicción de fallos, obteniéndose un 97% de exactitud (accuracy) de manera general en la aplicación del modelo inteligente al sistema. Se concluye que la aplicación del aprendizaje profundo es mucho más eficiente comparado con el aprendizaje automático clásico.

Palabras Clave: Inteligencia artificial, mantenimiento predictivo, Redes Neuronales Artificiales, Aprendizaje profundo.

 OPEN ACCESS



1. INTRODUCCIÓN

Los sistemas hidráulicos son sistemas amplios de utilización en la industria, ayudan a transportar cargas con esfuerzos pequeños (1), el uso incorrecto de manipulación de sistemas se ha visto afectado al momento de aplicar un correcto mantenimiento preventivo para la detección de fallas en los diferentes sistemas, incluidos sistemas hidráulicos; los cuales son utilizados en la industria moderna ya que desempeñan un papel insustituible en la mayor parte de los sistemas industriales, ocurriendo pérdidas económicas. Para garantizar que el sistema hidráulico tenga un desempeño normal, es necesario contar con un sistema de control del estado del sistema hidráulico fiable y preciso. Los métodos de monitorización de la condición que se basan en algoritmos clásicos de aprendizaje automático o aprendizaje profundo, por ejemplo: Redes neuronales (ANN), máquina de soporte vectorial (SVM) entre otros, sin embargo, cuando elegimos un método de clasificación o regresión de detección para aplicar a los sistemas hidráulicos, de manera general no se obtiene una alta precisión y es difícil elegir un algoritmo de alto rendimiento (2). Los estudios realizados con aplicación de redes neuronales son escasos, pero hoy en día se pueden implementar para la resolución de problemas en el campo de la ingeniería, con diseño de sistemas de control y optimización de procesos. El amplio uso de los sistemas es una gran mejora y simplifica los procesos en el mantenimiento a predecir fallos, monitoreando constantemente sistemas en la industria, no existen avances significativos en el uso de aplicación de inteligencia artificial para resolver problemas aplicados al campo de la ingeniería mecánica debido a la ausencia de la implementación (3).

2. METODOLOGÍA

La metodología se basa en la aplicación de Aprendizaje profundo (Deep Learning) para la detección de fallos en los sensores de un sistema hidráulico. Se utiliza datos previamente obtenidos del repositorio de la Universidad de California, Irvine. Dichos datos son de un banco de pruebas hidráulicos basados en múltiples sensores. con la aplicación del aprendizaje profundo se desea mejorar el estudio de detectar la vida de los sensores del sistema hidráulico, un estudio ya realizado por Guo 2019 (2), demostró la efectividad de predicción de fallos utilizando Aprendizaje automático (Machine Learning) con redes neuronales y varios algoritmos clásicos. Se pondrá a prueba los datos para plantear una estructura de análisis de datos y realizar un entrenamiento de datos previos a la aplicación de un modelo de Redes Neuronales Artificiales con aprendizaje

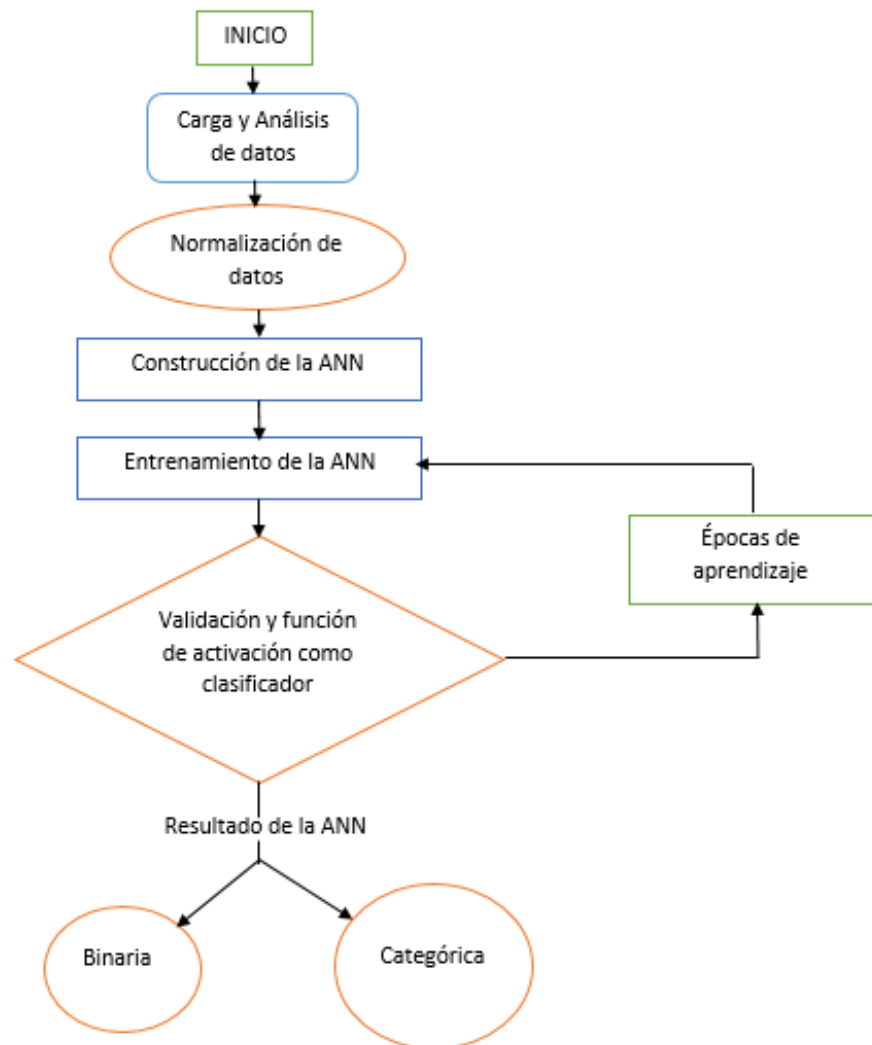


Figura 1

Flujograma para la aplicación del sistema inteligente. Realizado por: Mera H. 2022.

profundo siguiendo pasos previamente establecidos y planteados por el siguiente flujograma mostrados en la figura 1.

Se estable el proceso resumido del modelo inteligente en la Figura 1, utilizando redes neuronales artificiales como clasificador, utilizando como inicio el software libre Python (4), para el uso y ejecución de sistemas inteligentes basados en tensorflow (5), para la obtención y análisis de resultados, empleando pruebas de validación y funcionalidad a continuación se detalla todo el proceso de aplicación del modelo inteligente:

**Tabla 1**

DATOS Y CARACTERÍSTICAS GENERALES DEL SISTEMA HIDRÁULICO.

Sensor	Cantidad física	Unidad	Tasa de muestreo (Hz)	Atributos
PS1 - Sensor de presión	Presión	bar	100	6000
PS2 - Sensor de presión	Presión	bar	100	6000
PS3 - Sensor de presión	Presión	bar	100	6000
PS4 - Sensor de presión	Presión	bar	100	6000
PS5 - Sensor de presión	Presión	bar	100	6000
PS6 - Sensor de presión	Presión	bar	100	6000
EPS1 - Sensor de potencia del motor	Potencia de Motor	W	100	6000
FS1 - Sensor de caudal volumétrico	Volumen bajo caudal volumétrico	l/min	10	600
FS2 - Sensor de caudal volumétrico	Volumen bajo caudal volumétrico	l/min	10	600
TS1 - Sensor de temperatura	Temperatura	°C	1	60
TS2 - Sensor de temperatura	Temperatura	°C	1	60
TS3 - Sensor de temperatura	Temperatura	°C	1	60
TS4 - Sensor de temperatura	Temperatura	°C	1	60
VS1 - Sensor de vibración	Vibración	mm/s	1	60
CE - Sensor de eficiencia de enfriamiento virtual	Eficiencia de Enfriamiento (virtual)	%	1	60
CP - Sensor de potencia de refrigeración virtual	Poder de Enfriamiento (virtual)	kW	1	60
SE - Factor de eficiencia	Factor de eficiencia	%	1	60

Fuente: (Guo et al, 2019)
Realizado por: Mera H. 2022

2.1. Carga y análisis general de la base de datos

Se obtiene datos de Machine Learning Repository de la Universidad de California, Irvine. Los atributos son datos del sensor (todos numéricos y continuos) de mediciones

tomadas en el mismo momento, respectivamente, del ciclo de trabajo de un banco de pruebas hidráulico. Los sensores se leyeron con diferentes velocidades de muestreo, lo que dio lugar a diferentes números de atributos por sensor a pesar de que todos estaban expuestos al mismo ciclo de trabajo. El conjunto de datos contiene datos de sensor de proceso sin procesar (es decir, sin extracción de características) que están estructurados como matrices (delimitadas por tabulaciones) con las filas que representan los ciclos y las columnas los puntos de datos dentro de un ciclo (6). Como se los muestra a continuación:

Se realiza un análisis de parámetro de condición para la predicción de fallas las cuales se dividen en; condición de enfriamiento en %, condición de la válvula en %, fuga interna de la bomba, acumulador hidráulico en bar y la estabilidad, parámetros a medir cuando se realice la predicción de fallas según una condición específica.

Se utiliza librerías de Python básicas para la visualización de datos en forma de matrices (7), dichas librerías ya vienen preinstaladas para la manipulación y aplicación.

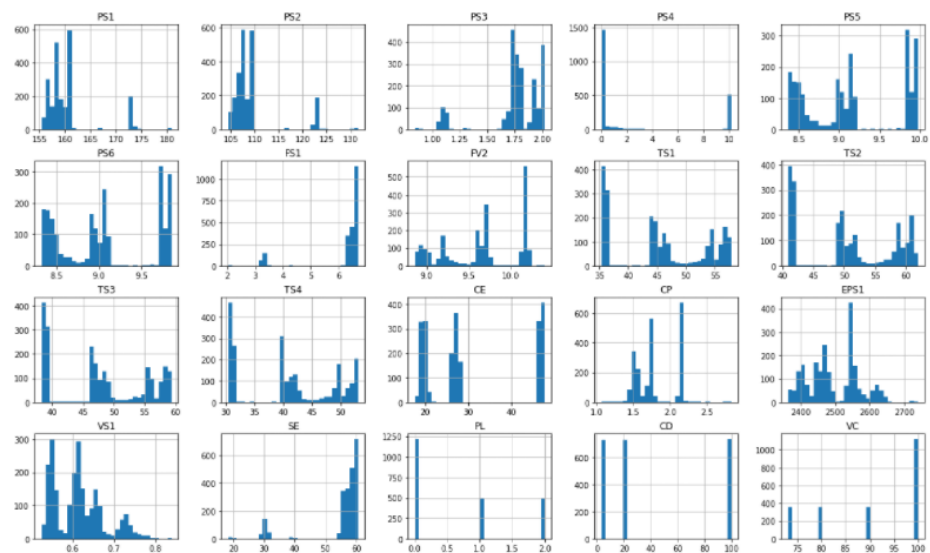


Figura 2

Histograma datos. Realizado por: Mera H. 2022.

Con la librería de matplotlib se grafica un histograma de distribución de datos, con la finalidad de observar el posicionamiento de la base de datos, por defecto sesgados a la derecha o izquierda o a su vez si cumplen una distribución como se observa en la figura 2.

Una vez realizado un análisis general, se aplica una correlación de datos para medir la cantidad de patrones que podrían existir previo a la aplicación del algoritmo inteligente.



Figura 3

Mapa de calor de correlación de datos. Realizado por: Mera H. 2022.

Con un mapa de calor, figura 3, se visualiza la cantidad de datos que son correlacionados de manera positiva y negativa para la cantidad de patrones que medirá el algoritmo inteligente.

2.2. Normalización de datos por condición

Hay que tomar en cuenta, en el Machine Learning (8) y Deep Learning (9), la cantidad de condiciones que se necesita analizar ya que se necesita realizar un análisis específico por condición, ya que dichos datos varían entre ellos y podría existir confusión al momento de aplicar los datos en la red neuronal (9).

A continuación se utiliza la librería de sklearn previamente instalada para usar el método de clasificación mutua, que ayuda con la medición de los datos específicos por condición, para la visualización de correlación de datos con cada uno de los sensores o características independientes como se observa a continuación:

A continuación, se utilizan los datos de etiqueta o datos de condición y se los transforma a binarios figura 5, con la función `get_dummies`, previo a la preparación de datos que se insertan en la red neuronal, ya que las condiciones son categóricas y poseen subcondiciones de análisis.



```

PS1    0.279429
PS2    0.269172
PS3    0.189773
PS4    0.099495
PS5    0.392179
PS6    0.390753
FS1    0.299570
FV2    0.294233
TS1    0.408156
TS2    0.374132
TS3    0.390146
TS4    0.392441
CE      0.196252
CP      0.123257
EPS1   0.271929
VS1    0.171554
SE      0.285827
dtype: float64
    
```

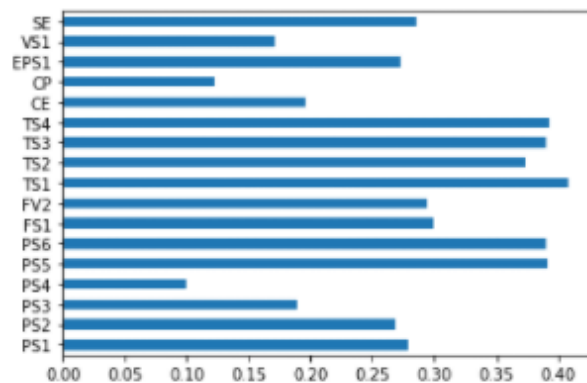


Figura 4

Correlación por condición. Realizado por: Mera H. 2022.

2.3. División de datos de entrenamiento, validación y prueba

Es muy importante considerar los datos propuestos, que se proponga un porcentaje de prueba y entrenamiento para que el modelo tenga la probabilidad de iniciar con el máximo error y terminar con el mínimo, es decir el entrenamiento del modelo ayuda a que pueda tener un desenvolvimiento previo antes de ser aplicado a la realidad, simulando la menor cantidad de errores cometidos en el aprendizaje del entrenamiento, en este caso se determinará con un 80:20 es decir la prueba se realiza primero un 20% de entrenamiento y 80% de testeo lo cual se tomara como guía para aplicar testeo, para alcanzar la mejor precisión posible del modelo aplicado a la detección de fallos en sistemas hidráulicos.

Se utiliza la librería de sklearn, la función `train_test_split` que automáticamente simula la declaración de las variables que serán sometidas al modelado de la red neuronal artificial, se declara la variable de la primera condición, que se aplica el mismo procedimiento para las demás condiciones de predicción, únicamente cambiando de



	73	80	90	100
0	0	0	0	1
1	0	0	0	1
2	0	0	0	1
3	0	0	0	1
4	0	0	0	1
...
2200	0	0	0	1
2201	0	0	0	1
2202	0	0	0	1
2203	0	0	0	1
2204	0	0	0	1

2205 rows × 4 columns

Figura 5

Transformación de datos a binarios. Realizado por: Mera H. 2022.

variables según la condición en este caso X e Y que se esté modelando. Siendo variables de entrenamiento (X_train) figura 6, y prueba (y_test) figura 7, utilizados para cuando se aplique el modelo de Deep Learning, además se observa que con la función test_size se denomina el porcentaje de entrenamiento del modelo y random_state la cantidad de semillas aleatorias que se crearán según el aprendizaje del algoritmo.

Para que exista un correcto análisis de datos, ya que no poseen el mismo rango de valores entre las variables X e Y, se necesitaría mucho cálculo y eso requiere una gran cantidad de tiempo, es decir para que éste inconveniente no afecte en el análisis de datos cuando se inserte en el modelo de inteligencia artificial, se realiza un escalado de datos, los valores de "X" se necesita que el rango entre valores leídos no sean demasiado amplios convirtiéndose en un rango particular normalizando todos los valores, utilizando la librería de sklearn, StandardScaler se los transforma de manera automática, preparando los datos para ser introducidos en la red neuronal.

El proceso de escalado se realiza para todas las condiciones de predicción sin excepción o para las condiciones que plantee el problema al cual se vaya aplicar una red neuronal para predecir de X variable.



datos de entrenamiento			PS1	PS2	PS3	PS4	PS5	PS6
572	156.259090	105.109720	1.686528	0.000000	8.409337	8.359900		
117	157.029422	106.361600	1.729071	0.000000	8.592279	8.533665		
1161	159.114915	107.267203	1.803048	0.000000	9.136608	9.051796		
789	158.176940	107.214707	1.786352	0.000000	8.970626	8.898110		
1322	158.989445	107.983810	1.806357	0.000158	9.146062	9.061882		
...		
1033	158.867803	107.103200	1.789223	0.000000	9.054415	8.976108		
1731	160.771032	109.054252	1.935512	10.112827	9.880726	9.761596		
763	157.909970	106.973936	1.779226	0.000000	8.924975	8.852112		
835	158.262197	107.266625	1.803428	0.000000	8.985091	8.909845		
1653	161.047945	109.579380	2.008050	0.785936	9.937681	9.819300		

	FS1	FV2	TS1	TS2	TS3	TS4
572	6.572717	8.932982	57.330050	61.493300	58.674333	52.562750
117	6.601323	9.239225	53.610750	58.093367	55.153417	49.000933
1161	6.632630	9.690735	44.593617	49.634633	46.804117	40.021017
789	6.634915	9.604517	47.109183	52.262500	49.325900	42.666800
1322	6.672445	9.723228	44.389533	49.417183	46.637700	39.878583
...
1033	6.643458	9.662503	45.757833	50.665533	47.919200	41.195717
1731	6.500505	10.175213	35.897400	41.499200	38.796267	30.888767
763	6.626108	9.544212	47.881850	53.009617	50.051850	43.319533
835	6.636710	9.607130	46.754067	51.858083	48.984550	42.363117
1653	6.687085	10.195733	35.514217	41.111717	38.441000	30.552333

	CE	CP	EPS1	VS1	SE
572	18.882850	1.471883	2367.347967	0.727367	59.950200
117	20.267500	1.531833	2405.333733	0.653417	60.262533
1161	27.786933	1.770233	2460.764700	0.612783	58.852000
789	26.399917	1.724267	2447.128100	0.597333	59.569083
1322	27.348517	1.772367	2464.307600	0.602217	59.857767
...
1033	28.058167	1.751483	2460.162400	0.616033	58.991450
1731	47.345067	2.168317	2553.740300	0.542833	56.692333
763	26.283400	1.731600	2436.215500	0.595333	59.623317
835	26.282333	1.713750	2443.892400	0.603817	59.742533
1653	47.129133	2.166700	2542.090533	0.562700	58.897267

Figura 6

Datos de entrenamiento. Realizado por: Mera H. 2022.

```

datos de prueba 572      0
117      1
1161     0
789      1
1322     0
..
1033     0
1731     0
763      1
835      1
1653     1
Name: Stable, Length: 1764, dtype: int64
    
```

Figura 7

Datos de prueba. Realizado por: Mera H. 2022.

2.4. Construcción de la Red neuronal artificial

En los sistemas hidráulicos la aplicación de Redes Neuronales Artificiales (ANN) se acoplan como herramienta de prevención y predicción, ya que los sistemas poseen

una compleja detección de fallas, que ocultan potentes señales no lineales que varían con el tiempo y el complejo mecanismo de transmisión de las vibraciones; por lo tanto aplicar la inteligencia artificial para la detección es una de las mejores alternativas para las industrias que mejorarán la productividad operacional de los equipos (10).

El modelo matemático consta de:

$$z = f (w_1x + w_2x + b) \quad (1)$$

X e Y son entradas

W_1 y W_2 son pesos sinápticos correspondientes a cada entrada

b es un término aditivo

f es una función de activación

z una salida

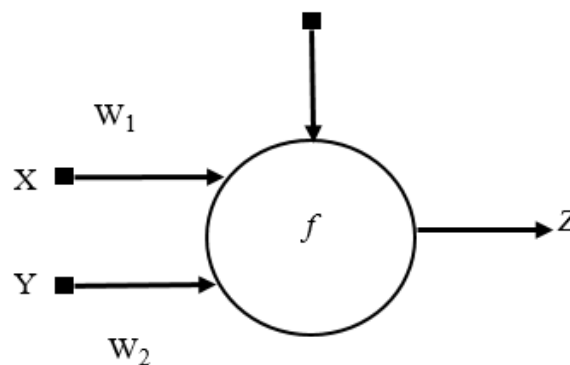


Figura 8

Red Neuronal Artificial. Fuente:Tablada -Germán y Torres, 2021.

Las entradas x e y son el estímulo que la neurona artificial recibe del entorno que la rodea, y la salida z es la respuesta a tal estímulo. La neurona se adapta al medio circundante y aprende de él, modificando el valor de sus pesos sinápticos w_1 y w_2 y su término aditivo b. Éstos son conocidos como los parámetros libres del modelo, pues los mismos pueden ser modificados y adaptados para realizar una tarea determinada (11).

2.5. Red neuronal artificial

Hay que tomar en cuenta el análisis el tipo de red neuronal ya que en el Deep Learning la extracción de características es binaria y la red necesita ser multicapa.

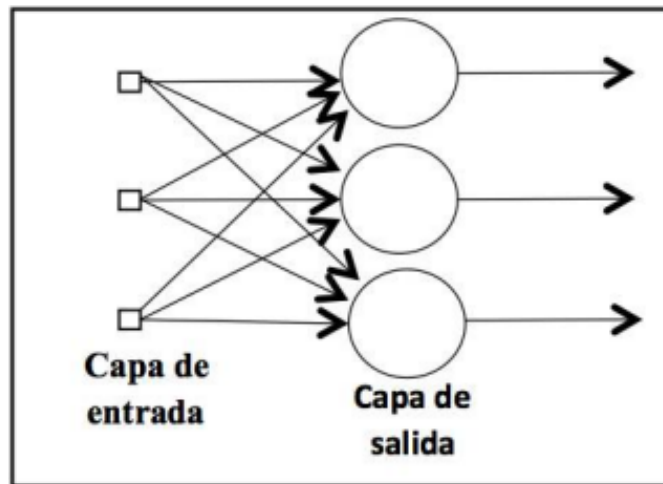


Figura 9

Red Neuronal Mono capa. Fuente: (Rivas, Bertha y Olivo, 2017).

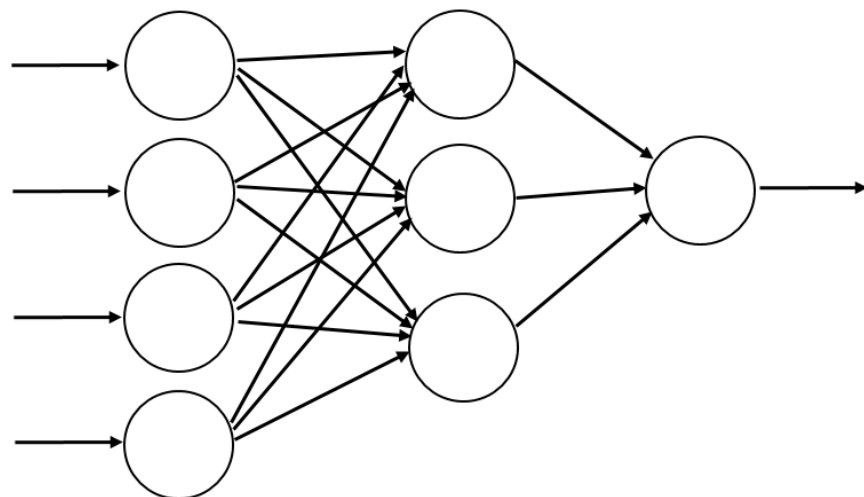


Figura 10

Red Neuronal Multicapa. Fuente:(Rivas, Bertha y Olivo, 2017).

La red neuronal de una sola capa, figura 9, es la más sencilla de todas ya que posee simplemente una sola capa de entrada lo cual su función es hacer una proyección hacia la capa de salida en la cual existen cálculos necesarios (12).

Es un complemento o aumento de la red mono capa, figura 10, existiendo un conjunto de capas entre la entrada y la salida que dichas capas, se las denomina capas ocultas, forman una secuencia en la entrada hasta la proyección de la salida mediante los cálculos que se efectúan a través de dicho proceso (12).



2.6. Construcción de la Red neuronal artificial

Se procede a construir la red neuronal una vez que se normalizaron y escalaron los datos, basados en tensorflow que crea, construye y ejecuta algoritmos de Machine Learning (5).

Se utiliza la librería de Theano que optimiza, define y evalúa las expresiones matemáticas que involucran matrices multidimensionales de manera eficiente, usándose como uno de los compiladores matemáticos de CPU (Unidad central de procesamiento) y GPU (Unidad de procesamiento gráfico) más utilizados en la comunidad Machine Learning (13).

Se aplica Keras biblioteca de Python de alto nivel que compacta y es sencillo de aprender para emplear en el aprendizaje que puede ejecutarse sobre TensorFlow (Theano o CNTK: kit de herramientas cognitivas de Microsoft). Esto permite a que los programadores se centren en los conceptos principales del Deep Learning, como la creación de capas en Redes Neuronales Artificiales existiendo dos tipos principales de estructuras: la API secuencial y la API funcional, el secuencial API se basa en la idea de una secuencia de capas; este es el más común. El modelo secuencial se considera como un grupo lineal de capas (14).

La librería keras importa métricas importantes cuando se construye la red neuronal, a continuación, se denominará las utilizadas:

Optimizador “Adam”: método que se basa en el descenso de gradiente estocástico que es la estimación que se adaptan momentos de primer y segundo orden (14).

Activador “Relu”: Selecciona la cantidad de neuronas ocultas para que no exista un sobreajuste, el activador no causa problema de gradiente de fuga, no se congestiona y es fácil de computar, evitando que el peso cambie su valor (15).

Activador “Sigmoid”: es una función que tiene una tasa de activación a las neuronas junto con “Relu”, dicha función se utiliza con categorías binarias de predicción (16).

Activador “Softmax”: utilizado generalmente para la clasificación de multicategorías en la capa de salida de la red neuronal (17).

```
# capa de entrada
red_neuronal.add(Dense(units=64, kernel_initializer='uniform', activation = 'relu', input_dim=17))
# 1era capa oculta
red_neuronal.add(Dense(units=64, kernel_initializer='uniform', activation = 'relu'))
#2da capa oculta
red_neuronal.add(Dense(units=32, kernel_initializer='uniform', activation = 'relu'))
#capa de salida
red_neuronal.add(Dense(units=1, kernel_initializer='uniform', activation = 'sigmoid'))
```

Figura 11

Red Neuronal con activador sigmoid. Realizado por: Mera H. 2022.



Como se observa en la figura 11, la red neuronal se aplica con un activador sigmoid ya que esta optimizado para una predicción binaria con salida 1, para realizar las predicciones de multicondición se cambia el activador sigmoid por softmax y las salidas con la cantidad de subcondiciones que posee la condición ya que softmax trabaja igual que sigmoid, pero esta optimizado para realizar una predicción de multinivel o multicondición, figura 12.

```
#capa de salida
red_neuronal.add(Dense(units=4, kernel_initializer='uniform', activation = 'softmax'))
```

Figura 12

Red Neuronal con activador softmax. Realizado por: Mera H. 2022.

“adam” método de tensorflow, es el optimizador que se basa o realiza el descenso de gradiente estocástico, éste optimizador analiza y actualiza los pesos de las entradas en el entrenamiento y reduce su perdida, usando la función `binary_crossentropy` se analiza de manera binaria la predicción, figura 13. Para la evaluación del modelo se usa la métrica de precisión “accuracy”.

```
#parametros de la capa
red_neuronal.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])
```

Figura 13

Compilación de la red.

En el análisis de categórico se reemplaza la función de `binary_crossentropy` por `categorical_crossentropy`, para que se analice varias respuestas en las capas de salidas permanece la métrica de la precisión

2.7. Ejecución y épocas de aprendizaje

Se denomina una historia con la función “fit” figura 14, la cual contiene todos los datos o repeticiones de entrenamiento que tendrá el modelo. Con `batch_size` compara la cantidad de lotes por segundo que realizara la red al momento de entrenamiento, el número de enteramiento o iteraciones que debe realizarse, se la plantea con la función `Epoch`.

El `Epoch` es el número de iteraciones en las cual realiza todo el proceso que se planteó en la construcción de la red que aplica métricas de análisis.

Al ejecutarse la historia inicia la ejecución de las iteraciones, figura15, según las épocas que se plantee el aprendizaje inicia con un máximo error medido por “loss”, ya que es el primer intento de aprendizaje de la red y mediante la métrica “accuracy” inicia con una baja precisión.



```
history= red_neuronal.fit(X_train, y_train, batch_size=200, epochs=2000)
```

Figura 14

Historia de la red. Realizado por: Mera H. 2022.

```
Epoch 1/4000
12/12 [=====] - 0s 21ms/step - loss: 0.6904 - accuracy: 0.6344 - val_loss: 0.6832 - val_accuracy: 0.67
80
Epoch 2/4000
12/12 [=====] - 0s 3ms/step - loss: 0.6693 - accuracy: 0.6519 - val_loss: 0.6320 - val_accuracy: 0.678
0
Epoch 3/4000
12/12 [=====] - 0s 4ms/step - loss: 0.6093 - accuracy: 0.6519 - val_loss: 0.5732 - val_accuracy: 0.678
0
Epoch 4/4000
12/12 [=====] - 0s 4ms/step - loss: 0.5720 - accuracy: 0.6519 - val_loss: 0.5445 - val_accuracy: 0.684
8
Epoch 5/4000
12/12 [=====] - 0s 4ms/step - loss: 0.5273 - accuracy: 0.6859 - val_loss: 0.4914 - val_accuracy: 0.768
7
Epoch 6/4000
12/12 [=====] - 0s 4ms/step - loss: 0.4745 - accuracy: 0.8129 - val_loss: 0.4352 - val_accuracy: 0.786
8
Epoch 7/4000
12/12 [=====] - 0s 4ms/step - loss: 0.4220 - accuracy: 0.8464 - val_loss: 0.3766 - val_accuracy: 0.868
5
Epoch 8/4000
12/12 [=====] - 0s 4ms/step - loss: 0.3639 - accuracy: 0.8838 - val_loss: 0.3353 - val_accuracy: 0.895
7
Epoch 9/4000
12/12 [=====] - 0s 4ms/step - loss: 0.3264 - accuracy: 0.8980 - val_loss: 0.3083 - val_accuracy: 0.891
2
```

Figura 15

Ejecución y aprendizaje de la red neuronal. Realizado por: Mera H. 2022.

Una vez terminado el aprendizaje se mide el Overfitting y Underfitting aparece finalmente cuando se habla del grado polinomial, dicho grado se representa cuanta flexibilidad en el modelo de machine learning existe, con una potencia alta que permite que el modelo tenga facilidad para alcanzar los puntos de datos como sea lo posible (18).

Overfitting y Underfitting son constantes y omnipresentes en el aprendizaje automático, el objetivo del aprendizaje es aproximar o ajustar señales verdaderas que relacionan las características X con su respectiva propuesta en Y, que se pueda interpretar como una función $f: X \rightarrow Y$ o una distribución $P(Y | X)$ (19).

Se mide con un gráfico los valores perdidos reales con los valores perdidos virtuales para la visualización de un desenvolvimiento optimo del aprendizaje de la red previo a la predicción de resultados.

3. RESULTADOS

Los resultados son planteados en una matriz de confusión, se aplica después del entrenamiento de los datos que se utilizaron en el algoritmo de inteligencia artificial para demostrar la evaluación utilizada en el Machine Learning o Deep Learning. Una matriz de confusión o una matriz de error, crea varias predicciones y resultados de

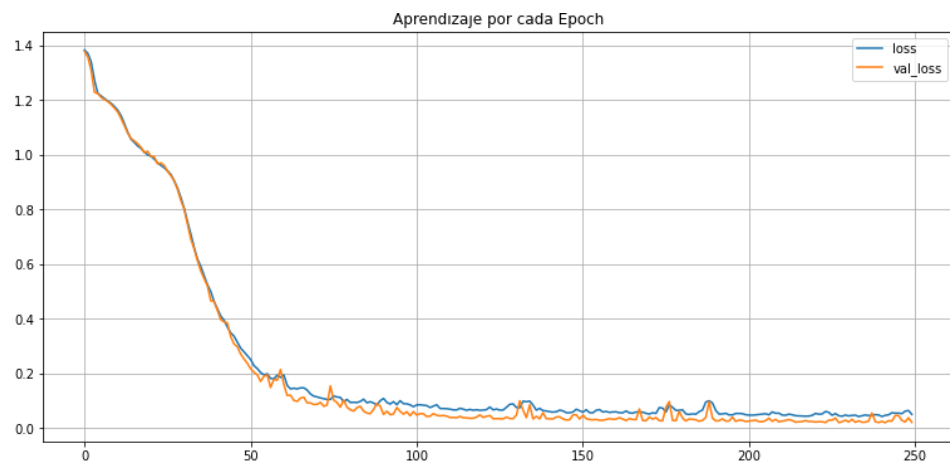


Figura 16

Comprobación de la red. Realizado por: Mera H. 2022.

prueba figura 17, también ayuda a obtener un análisis en profundidad de los datos estadísticos más rápido a través de una visualización de datos (20).

Las columnas de una Matriz de confusión son los resultados del tipo de predicción y las filas los resultados reales, son enumerados todos los resultados posibles, tomando el problema como una clasificación binaria (21).

La matriz de confusión es una tabla de frecuencias bidireccionales figura 17, con dos variables binarias reales (bueno o malo) y predicho (bueno o malo) los valores son cuatro elementos que se denominan: Verdadero Negativo (TN), Verdadero Positivo (TP), Falso Negativo (FN) y Falso Positivo (FP), no necesariamente deben ir en orden.

Verdadero negativo es el número de negativos (reales) que se clasifican correctamente como negativos.

Falso negativo es el número de positivos (reales) que se clasifican incorrectamente como negativos.

Verdadero positivo es el número de positivos (reales) que se clasifican correctamente como positivos

Falso positivo es el número de negativos (reales) que se clasifican incorrectamente como positivos.

Desde el punto de vista de la lingüística, podemos interpretar Falso como "no actual" (22). Con la matriz de confusión se calculan las métricas de análisis con las siguientes formulas:



Figura 17

Matriz de confusión. Fuente: Babel, Kumar Singh y Kumar Jangir, 2019.

$$\text{Precisión} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

$$F1 \text{ Score} = 2 \left(\frac{\text{Precisión} \times \text{Recall}}{\text{Precisión} + \text{Recall}} \right) \quad (4)$$

De manera manual se pueden obtener los valores de precisión, recall y f1-score con las formulas anterior mencionadas, o su vez Python ofrece un reporte de clasificación que muestra automáticamente todos los resultados de la predicción.

3.1. Resultado de la Estabilidad

La estabilidad es una condición general del sistema hidráulico, que demuestra una predicción binaria de 0 cuando el sistema no posee ninguna falla en su sistema y 1 cuando su sistema como se muestra en la figura 18, la matriz de confusión con su respectivo reporte de clasificación que demuestran la efectividad del sistema en la matriz de confusión de 2 por 2 se observa que demuestra el desenvolvimiento que tuvo la aplicación de la red neuronal para la predicción de fallos, la suma total de sus datos de testeo son 441, mostrados en la tabla de resultados, separados de la siguiente manera, 296 puntos en predicciones verdaderas cuando la condición de estabilidad esta en 0 y 133 puntos de predicciones preventivas cuando está en fallo o 1.



A continuación, con la matriz de confusión de la estabilidad se procede a calcular las métricas que miden la eficiencia de la red neuronal, las métricas nombradas a continuación se las calcula con las formulas mencionadas anteriormente, se inicia con la precisión en la condición 0 se obtiene un 97% y en la condición 1 un 98% aquí nos damos cuenta que el sistema si posee un equilibrio, se procede a calcular el recall, en 0 un 99% y en 1 un 94% aquí vemos que varía por un 6%, se puede decir que la red posee caídas en neuronas irrelevantes y elimina valores, siendo un poco la variación en el recall, utilizando las métricas de la precisión y el recall se procede a calcular el f1-score quien predominara para la factibilidad de detección de fallos, en 0 posee un 98 y en 1 un 96%, es decir que tiene un desenvolvimiento optimo la red neuronal para predecir fallos y por último se demuestra el accuracy o exactitud es del 97% de la red neuronal aplicado a esta condición para predicción de fallos de la estabilidad.

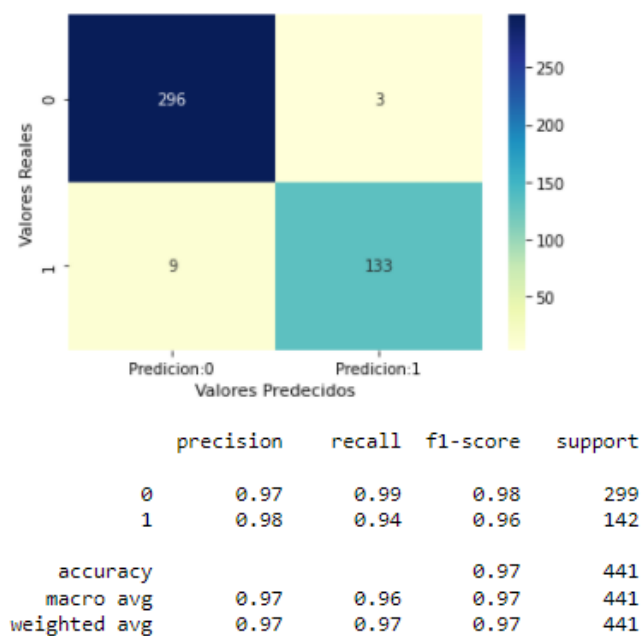


Figura 18

Resultado de la estabilidad. Realizado por: Mera H. 2022.

3.2. Resultado de la Condición de enfriamiento

La segunda condición de análisis de resultados es la condición de enfriamiento se basa en una predicción categórica formando una matriz de confusión de tres por tres junto con su reporte de clasificación, que posee tres subcondiciones las cuales son: al 3% cerca del fallo total, al 20% eficiencia reducida y al 100% full eficiencia, como se observa en la figura 19, en éste caso la eficiencia de la predicción de la matriz es excelente sin

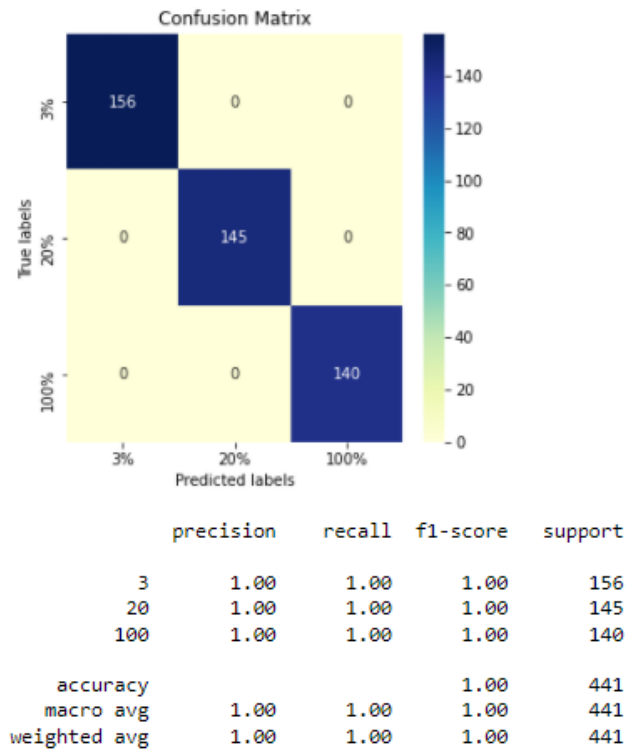


Figura 19

Resultado de la Condición de enfriamiento. Realizado por: Mera H. 2022.

existir valores falsos en sus intersecciones, es decir que para el 3% tenemos 156 valores para el 20% tenemos 145 valores y para el 100% tenemos 140 predichos sumando 441 datos de testeo con una exactitud del 100%, se calcula la precisión y como se observa en la tabla de reporte de métricas, la precisión en las 3 condiciones es el 100% en el recall de igual manera el 100% y usando estas 2 métricas se calcula el f1-score dándose el 100%, la exactitud de la red o accuracy es del 100%, demostrando que la red neuronal tiene un desempeño excelente en la detección de fallos del enfriamiento del sistema.

3.3. Resultado de la Condición de la válvula

La tercera condición de análisis es la condición de la válvula poseyendo una matriz de predicción categórica de cuatro por cuatro junto con su reporte de clasificación ya que posee cuatro subcondiciones: 73% cerca del fallo total, 80% severo retraso, 90% pequeño retraso y 100% comportamiento optimo, como se observa la figura 20, se mira un falso positivo en la condición del 90% en conjunto con la 80% es decir que existe un valor falso positivo que se lo debe analizar en una sola condición para descartarlo en el análisis de la otra condición y se calcula, para la precisión observando el reporte de clasificación, solo la condición del 80% tiene un 99% y los demás 100%, para el recall

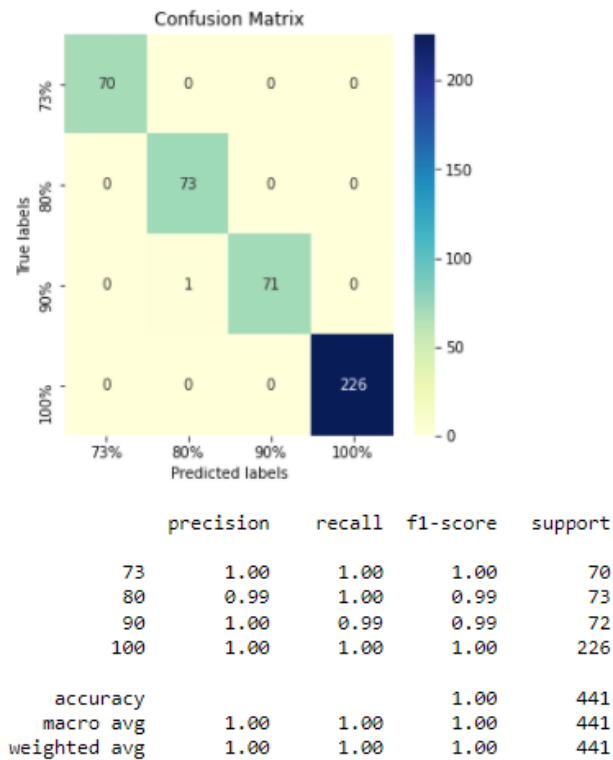


Figura 20

Resultado de la Condición de la válvula. Realizado por: Mera H. 2022.

solo el 99% posee al 90%, se calcula el f1-score y se plantea de la siguiente manera 100% para cuando la condición de la válvula está al 73% y 100% y 99% cuando esta al 80% y 90% y por último la exactitud o accuracy de la red es del 100%, demostrando que la red posee un desempeño alto para detectar fallos en la válvula.

3.4. Resultado de la Fuga interna de la válvula

La cuarta condición de análisis es la fuga interna de la bomba poseyendo una matriz de confusión categórica de tres por tres junto con el reporte de clasificación ya que posee tres subcondiciones, 0: ninguna fuga, 1: fuga de débil, 2: fuga severa, como se observa en la figura 21, la matriz posee varios puntos de predicción de falsos positivos, en especial para la subcondición 1 que posee 6 puntos a la derecha y 1 hacia arriba eso quiere decir que se asemejan a la subcondición 0, tomándolo como una alerta, en la condición 0 con 251 puntos, condición 1 con 95 puntos y 86 puntos de predicción reales o verdaderos sumando 441 datos usados para el testeo del modelo inteligente. En la subcondición 0 posee un 100% en la precisión, recall y f1-score es decir que posee una predicción excelente, en cambio para la subcondición 1 y 2 los porcentajes varían, ya que los falsos positivos en la matriz se asemejan entre valores de condición es decir



que están equilibrados por condición, en la condición 1: 99% precisión, 94% recall y 96% f1-score; en la condición 2: 99% precisión, 94% recall y 96% f1-score y por último el accuracy de la red es del 100%, demostrando que la red posee un desempeño alto para detectar fallos en la bomba del sistema.

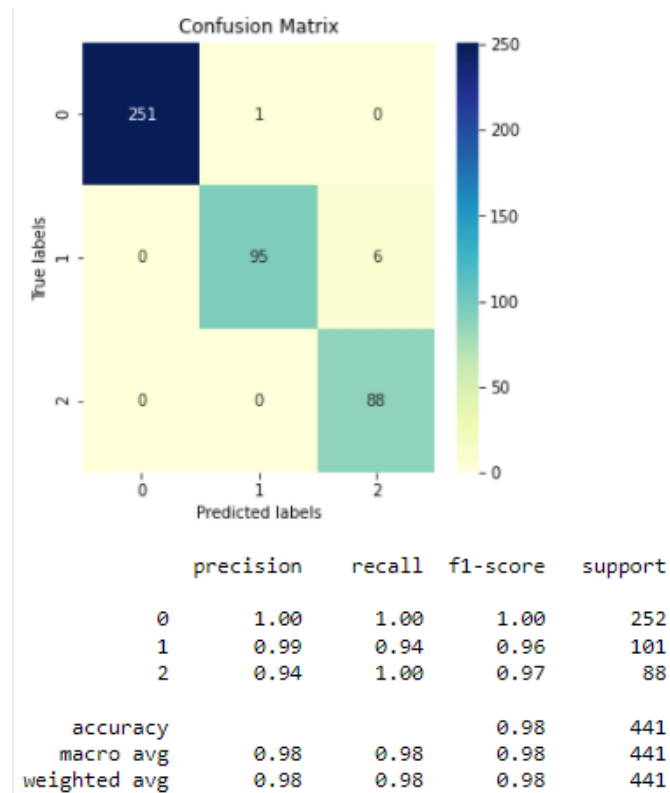


Figura 21

Resultado de la Fuga interna de la bomba. Realizado por: Mera H. 2022.

3.5. Resultado del Acumulador hidráulico

La quinta condición de análisis es la condición del acumulador hidráulico poseyendo una matriz de predicción categórica de cuatro por cuatro junto con su reporte de clasificación, ya que posee cuatro subcondiciones: 90 bar cerca de la falla total, 100 bar presión severamente reducida, 115 bar presión ligeramente reducida y 130 bar presión optima, como se observa la figura 22, existen varios falsos positivos en cada condición en todas las condiciones: para 90 bar de presión existe 146 puntos de predicción verdaderos y 7 falsos positivos con una precisión del 94%, un recall del 95% y un f1-score del 95% , para 100 bar de presión existe 68 puntos de predicción verdaderos y 20 falsos positivos con una precisión del 94%, un recall del 95% y un f1-score del 95%, para 115 bar de presión existe 70 puntos de predicción verdaderos y 3 falsos positivos con



una precisión del 80%, un recall del 96% y un f1-score del 88%, para 130 bar de presión existe 123 puntos de predicción verdaderos y 4 falsos positivos con una precisión del 98%, un recall del 97% y un f1-score del 97%, sumando todos los valores verdaderos con un total de 407 de los 441 ingresados con una accuracy o exactitud de predicción del 92% demostrando que la red posee un desempeño bueno para detectar fallos en el acumulador hidráulico.

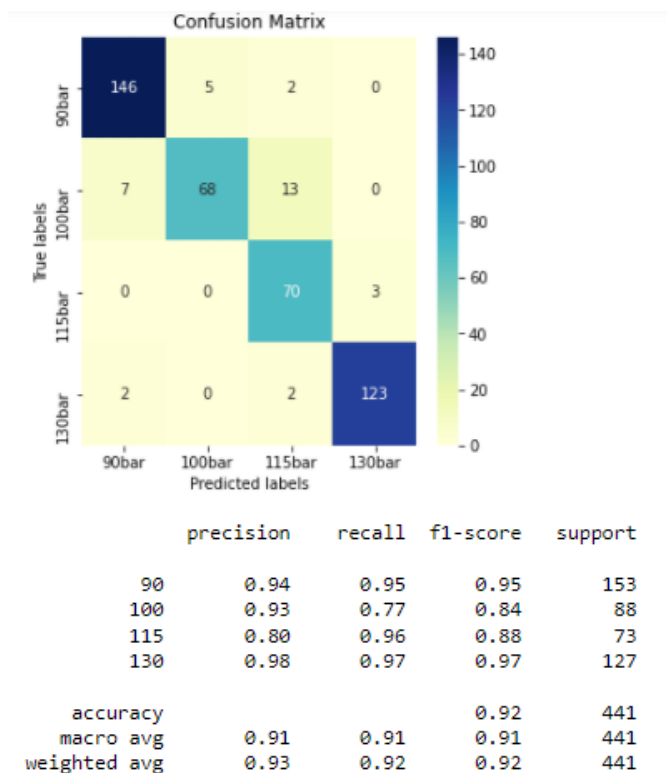


Figura 22

Resultado del acumulador hidráulico. Realizado por: Mera H. 2022.

3.6. Resultados generales

Según la tabla II, la exactitud demuestra como fue el aprendizaje de la red neuronal en la aplicación del modelo para el entrenamiento de predicción de valores aleatorios, es decir por cada condición la Red muestra cómo se desenvuelve y el nivel de Épocas o Iteraciones de aprendizaje que necesitan para que la exactitud de la red sea aceptable y poder llegar a la predicción con un total del 97% en todo el sistema hidráulico.

El f1-score demuestra la aceptación y credibilidad que posee el modelo inteligente para la detección de fallos en un sistema hidráulico, con se observa en la tabla II,

**Tabla 2**

TABLA DE RESULTADOS GENERALES EN DETECCIÓN DE FALLAS DE SISTEMAS HIDRÁULICOS.

Condiciones	Accuracy (%)	F1-score (%)	Épocas
Estabilidad	97	Estable: 98 No estable: 96	4000
Condición de enfriamiento	100	Full eficiencia: 100 Eficiencia reducida: 100 Cerca del fallo total: 100	500
Condición de la válvula	100	Comportamiento optimo: 100 Pequeño retraso: 99 Severo retraso: 99 Cerca total del fallo: 100	500
Fuga interna de la bomba	100	Ninguna fuga: 100 Fuga débil: 96 Fuga severa: 97	500
Acumulador hidráulico	88	Presión optima: 97 Presión ligeramente reducida: 88 Presión severamente reducida: 84 Cerca de la falla total: 95	1500
Resultados finales en la predicción de fallas del sistema hidráulico	97		

Realizado por: Mera H. 2022

posee un resumen general de los resultados obtenidos a través de la investigación, demostrando porcentajes elevados en la predicción de datos por cada subcondición.

Se compara los resultados ya obtenidos en el estudio realizado de Guo 2019 (2) que posee un 82.6% de media aplicando redes neuronales artificiales con aprendizaje automático, con el obtenido en la tabla II con un 97% de media

4. CONCLUSIONES

Se realizó un nuevo método de estudio para el monitoreo de condición en sistemas hidráulicos. Los datos son obtenidos del repositorio de la universidad de california (UCI), construyendo una red neuronal de tipo dense basado en aprendizaje profundo con un resultado del 97% de media en la exactitud para la detección de fallos en los sensores. Comparando con el estudio ya realizado de GUO 2019 (2) que aplica redes neuronales artificiales con aprendizaje automático (Machine Learning), SVM y LDA de métodos clásicos, se demostró que el estudio realizado con aprendizaje profundo tiene un mejor desempeño para la detección de salud de los sensores de un sistema hidráulico.



References

- [1] Prakash J, Kankar PK. Health prediction of hydraulic cooling circuit using deep neural network with ensemble feature ranking technique [Internet]. Measurement. 2020 Feb;151:107225. [cited 2021 Aug 30] Available from: <https://www.sciencedirect.com/science/article/abs/pii/S0263224119310905>
- [2] Guo P, Wu J, Xu X, Cheng Y, Wang Y. Health condition monitoring of hydraulic system based on ensemble support vector machine. 2019 Progn Syst Heal Manag Conf PHM-Qingdao 2019 [Internet]. 2019 Oct 1 [cited 2021 Nov 25]; Available from: <https://ieeexplore.ieee.org/abstract/document/8942981> <https://doi.org/10.1109/PHM-Qingdao46334.2019.8942981>
- [3] Pavlenko I, Trojanowska J, Ivanov V, Liaposhchenko O. Parameter identification of hydro-mechanical. Int J Mechatronics Appl Mech. 2018;(5):19–26.
- [4] Zadka M. Installing python. DevOps in Python [Internet]. 2019 [cited 2021 Nov 25];1–6. Available from: https://link.springer.com/chapter/10.1007/978-1-4842-4433-3_1 https://doi.org/10.1007/978-1-4842-4433-3_1.
- [5] MLSys Proceedings. TensorFlow.js: Machine learning for the web and beyond [Internet]. Machine Learning for the Web and Beyond. 2022 [cited 2022 Feb 28]. Available from: <https://proceedings.mlsys.org/paper/2019/hash/1d7f7abc18fcb43975065399b0d1e48e-Abstract.html>
- [6] Dua D and G. © UCI Machine Learning Repository [Internet]. Condition monitoring of hydraulic systems Data Set. 2017 [cited 2021 Dec 17]. Available from: <https://archive.ics.uci.edu/ml/datasets/Condition+monitoring+of+hydraulic+systems>
- [7] Rolon-Mérette D, Ross M, Rolon-Mérette T, Church K. Python for research in psychology introduction to Anaconda and Python: Installation and setup. 2020 [cited 2021 Nov 22];16(5). Available from: <https://www.spyder-ide.org/>
- [8] Zhang XD. Machine Learning. A Matrix Algebr approach to Artif Intell [Internet]. 2020 [cited 2021 Nov 29];223–440. Available from: https://link.springer.com/chapter/10.1007/978-981-15-2770-8_6
- [9] Hao X, Zhang G, Ma S. Deep Learning. [Internet]. 2016 Nov 30 [cited 2022 Feb 28];10(3):417–439. Available from: <https://www.worldscientific.com/doi/abs/10.1142/S1793351X16500045> <https://doi.org/10.1142/S1793351X16500045>
- [10] Dai J, Tang J, Huang S, Wang Y. Signal-based intelligent hydraulic fault diagnosis methods: Review and prospects. Chinese J Mech Eng (English Ed [Internet]. 2019;32(1). Available from: <https://doi.org/10.1186/s10033-019-0388-9>



- [11] Tablada -Germán CJ, Torres A. Redes Neuronales Artificiales. 2021 Aug 10 [cited 2021 Dec 16]; Available from: <https://revistas.unc.edu.ar/index.php/REM/article/view/10280>
- [12] Rivas W, Bertha A, Olivo M. Redes neuronales artificiales aplicadas al reconocimiento de patrones. 2017 [cited 2022 Feb 3]; Available from: www.utmachala.edu.ec
- [13] The Theano Development Team. Al-Rfou R, Alain G, Almahairi A, Angermueller C, Bahdanau D, et al. Theano: A Python framework for fast computation of mathematical expressions. 2016 May 9 [cited 2022 Feb 28]; Available from: <http://arxiv.org/abs/1605.02688>
- [14] Manaswi NK. Understanding and working with Keras. Deep Learn with Appl Using Python [Internet]. 2018 [cited 2022 Feb 28];31–43. Available from: https://link.springer.com/chapter/10.1007/978-1-4842-3516-4_2
https://doi.org/10.1007/978-1-4842-3516-4_2
- [15] Alhassan AM, Zainon WM. Brain tumor classification in magnetic resonance image using hard swish-based RELU activation function-convolutional neural network. Neural Comput Appl 2021 3315 [Internet]. 2021 Jan 11 [cited 2022 Mar 2];33(15):9075–9087. Available from: <https://link.springer.com/article/10.1007/s00521-020-05671-3>
<https://doi.org/10.1007/s00521-020-05671-3>
- [16] Pan SP, Li ZF, Huang YJ, Lin WC. FPGA realization of activation function for neural network. Proc - 2018 7th Int Symp Next-Generation Electron ISNE 2018 [Internet]. 2018 Jun 22 [cited 2022 Mar 2];1–2. Available from: <https://ieeexplore.ieee.org/abstract/document/8394695>
<https://doi.org/10.1109/ISNE.2018.8394695>
- [17] Wei Z, Arora A, Patel P, John L. Design space exploration for softmax implementations. Proc Int Conf Appl Syst Archit Process [Internet]. 2020 Jul 1 [cited 2022 Mar 2];2020-July:45–52. Available from: <https://ieeexplore.ieee.org/abstract/document/9153236>
<https://doi.org/10.1109/ASAP49362.2020.00017>
- [18] Follow WK. Overrtting vs. Underrtting: A complete example. 2018 [cited 2021 Dec 21]; Available from: <https://towardsdatascience.com/overfitting-vs-underfitting-a-complete-example-d05dd7e19765>
- [19] Bashir D, Monta nez GD, Sehra S, Segura PS, Lauw J. An Information-theoretic perspective on overfitting and underfitting. Lect Notes Comput Sci (including Subser Lect Notes Artif Intell Lect Notes Bioinformatics) [Internet]. 2020 Nov 29 [cited 2021 Dec 21];12576 LNAI:347–58. Available from: https://link.springer.com/chapter/10.1007/978-3-030-64984-5_27
https://doi.org/10.1007/978-3-030-64984-5_27



- [20] Babel V, Kumar Singh B, Kumar Jangir S. Journal of Analysis and Computation (JAC) Evaluation Methods for Machine Learning. 2019 [cited 2021 Dec 15]; Available from: www.ijaonline.com
- [21] Xu J, Zhang Y, Miao D. Three-way confusion matrix for classification: A measure driven view. Inf Sci (Ny) [Internet]. 2020 Jan 1 [cited 2021 Dec 16];507:772–94. Available from: <https://www.sciencedirect.com/science/article/abs/pii/S0020025519306024>
<https://doi.org/10.1016/j.ins.2019.06.064>.
- [22] Zeng G. On the confusion matrix in credit scoring and its analytical properties. <https://doi.org/https://doi.org/101080/0361092620191568485>. [Internet]. 2019 May 2 [cited 2021 Dec 17];49(9):2080–93. Available from: <https://www.tandfonline.com/doi/abs/10.1080/03610926.2019.1568485r-limb-prosthesis>. RIAI - Revista Iberoamericana de automática e informática Industrial [Internet]. 2008 [cited 2023 Jan 22];5(2):60–68. Available from: <http://riai.isa.upv.es>
- [23] Rosales, Daniela (INSHT) IN de S e H en el T. Evaluación de las condiciones de trabajo: carga postural. Método REBA (Rapid Entire Body Assessment). Instituto Nacional de Seguridad e Higiene en el trabajo [Internet]. 2001 [cited 2023 Jan 22];7. Available from: http://www.insht.es/InshtWeb/Contenidos/Documentacion/FichasTecnicas/NTP/Ficheros/601a700/ntp_601.pdf
- [24] Boné Pina MJ. Método de evaluación ergonómica de tareas repetitivas, basado en simulación dinámica de esfuerzos con modelos humanos. 2016 [cited 2023 Jan 22];268. Available from: <https://dialnet.unirioja.es/servlet/tesis?codigo=78749&info=resumen&idioma=SPA>
- [25] Asensio cuesta S, Bastante Ceca M, Diego Mas J. EVALUACIÓN ERGONOMICA DE PUESTOS DE TRABAJO - ASENSIO CUESTA, SABINA, BASTANTE CECA, MARÍA JOSÉ, DIEGO MAS, JOSÉ ANTONIO - Google Libros [Internet]. 2012 [cited 2023 Jan 22]. p. 1–350. Available from: <https://books.google.com.ec/books?hl=es&lr=&id=v5kFfWOUh5oC&oi=fnd&pg=PR15&dq=1.+Asensio-Cuesta+S,+María+José+Bastante+Ceca,+Diego+A.+EVALUACIÓN+ERGONOMICA+DE+PUESTOS+DE+TRABAJO.+Editorial+Paraninfo%3B+2012&ots=wJVRmKqsFL&sig=Zzl4RNRdOO5Zwb11n4DXijNbeUE#v>
- [26] Guerrero Silva CF. Universidad de Guayaquil. Facultad de Ingeniería Industrial. Carrera de Ingeniería Industrial. Universidad de Guayaquil. Facultad de Ingeniería Industrial. Carrera de Ingeniería Industrial.; 2019 [cited 2023 Jan 22]. p. 72 Repositorio Universidad de Guayaquil: Evaluación de riesgo ergonómico aplicando el método REBA a los trabajadores administrativos de la Empresa



Pública Municipal Registro de la Propiedad de Guayaquil. Available from:
<http://repositorio.ug.edu.ec/handle/redug/42144>