

Intelligent Fault Detection through Real-Time Data Analysis *

Jesus G. Alvarez * Oscar D. Sanchez * Alma Y. Alanis**

* CUCEI, Universidad de Guadalajara, Blvd. Marcelino García
Barragán 1421. Col. Olímpica, Guadalajara, Jalisco (e-mail:
alma.alanis@academicos.udg.mx)

Abstract: This paper presents a comparative analysis of four strategies for intelligent fault classification based on real-time data analysis. The number of research publications proposing different intelligent classification strategies is extensive, but the vast majority of the proposed classifiers operate offline. This paper explores different intelligent classification techniques with the goal of implementing them in real-time for fault detection in nonlinear dynamic systems. Real-time data classification is highly relevant for the design of fault-tolerant control systems.

Keywords: Time series modeling; Fault detection and diagnosis; Artificial neural networks; Classification; Real-time systems.

1. INTRODUCCIÓN

La detección y aislamiento de fallas (DAF) es un tema de interés actual debido que proporciona la información necesaria para ajustar y mitigar los efectos de la falla. Generalmente, las fallas más comunes ocurren en los actuadores o en los sensores, por lo que este artículo propone un novedoso esquema libre de modelo para la detección y aislamiento de fallas en sensores y actuadores de motores de inducción (IM). La metodología propuesta tiene la tarea de detectar y aislar fallas a partir de la medición de datos en tiempo real para diferentes tipos de fallas en sensores y actuadores de motores de inducción (IM), ya sea por desconexión, degradación, falla o daño en la conexión o comunicación. El enfoque propuesto incluye el uso de redes neuronales profundas que no requieren un modelo nominal ni generan residuos para la detección de fallas, lo que las convierte en una herramienta útil y en tiempo real. Además, el enfoque de aislamiento de fallas se lleva a cabo mediante clasificadores que diferencian características independientemente de los demás clasificadores. Para esta tarea se utilizan la red neuronal de memoria a largo plazo (LSTM), la LSTM bidireccional (BiLSTM), el perceptrón multicapa (MLP) y la red neuronal convolucional (CNN). El esquema de detección y aislamiento de fallas propuesto para sensores y actuadores es simple. Puede aplicarse de forma conjunta con diversas aplicaciones que involucren esquemas de DAF. Los resultados muestran que las redes neuronales profundas son una herramienta potente y versátil para la detección y aislamiento de fallas a partir de datos en tiempo real Sanchez *et al.* (2022).

Este trabajo aborda el problema de la identificación y detección de fallas existentes, también denominada problema de diagnóstico. Este problema se divide en dos etapas: la de detección de fallas y la de aislamiento de fallas. En la primera, se determina si se ha producido una falla,

además de identificar el momento en que el sistema la ha sufrido Rodriguez-Guerra *et al.* (2019). Por otro lado, la etapa de aislamiento de fallas se centra en determinar el componente donde se originó la falla y realizar acciones para aislar la misma y evitar efectos adversos provocados por la falla presentada Rodriguez-Guerra *et al.* (2019).

Existen tres tipos principales de fallas en los motores de inducción: fallas del actuador, fallas de la planta y fallas del sensor. En aplicaciones industriales, el control tolerante a fallas ante fallas del sensor es crucial, ya que estos permiten el monitoreo de las variables de estado del sistema en algunos de los componentes con mayor potencial de falla. De hecho, como se menciona en Chen *et al.* (2020) que analizar las observaciones de los sensores es una tarea compleja. Además, los actuadores representan una conexión entre el algoritmo de control y el sistema, por lo que si hay una falla en el actuador, su capacidad operativa se reduce en comparación con el funcionamiento normal. Una de las principales aplicaciones de la detección y el aislamiento de fallas es garantizar la confiabilidad y la seguridad, como en el caso de los trenes de alta velocidad Chen *et al.* (2020).

Como se menciona en el trabajo publicado en Saufi *et al.* (2019), el enfoque tradicional para detectar y diagnosticar fallas se lleva a cabo en cinco pasos: adquisición de datos, procesamiento de datos, extracción de características, reducción dimensional y clasificación. Esto implica que encontrar las técnicas adecuadas en cada paso es fundamental y que es necesario un proceso de prueba y error. Por ello, diferentes trabajos utilizan enfoques únicos para abordar cada aplicación específica Sanchez *et al.* (2022).

En general, los trabajos existentes en clasificación trabajan fuera de línea. En Chen *et al.* (2020), se presenta una recopilación de numerosos artículos con diferentes metodologías de diagnóstico y detección, entre las que se mencionan métodos basados en: análisis de señales, modelos y basados en datos o sin modelos. Como se men-

* Los autores agradecen al apoyo otorgado por el CUCEI de la Universidad de Guadalajara para el desarrollo de esta investigación.

ción en dicho artículo, las estrategias desarrolladas para el uso de controladores tolerantes a fallas son un tema de interés para los investigadores; sin embargo, no existen muchos trabajos relacionados con esquemas de DAF para su aplicación en controladores.

Además, para aplicaciones en tiempo real, es necesario contar con un clasificador confiable que funcione en tiempo real, junto a un controlador tolerante a fallas. Este artículo propone una metodología para el diagnóstico en línea de fallas de actuadores y sensores con aplicación a motores de inducción trifásicos, utilizando únicamente datos medidos sin conocer el modelo nominal ni otras suposiciones previas, salvo los datos con los que se entrenó la RNA. Esta metodología se implementó con cuatro estructuras de redes neuronales profundas y sus rendimientos se probaron con datos experimentales en tiempo real.

2. REDES NEURONALES PROFUNDAS

Para aplicaciones reales, es necesario desarrollar métodos de diagnóstico de fallas capaces de analizar grandes cantidades de datos para su detección automática con precisión y rapidez. Con frecuencia, la detección de fallas en los motores de inducción, se basa en observadores; sin embargo, su principal desventaja reside en su gran dependencia de modelos matemáticos precisos correspondientes al sistema. Esto no es completamente efectivo en aplicaciones reales porque los parámetros del sistema suelen variar durante el proceso y las perturbaciones son desconocidas, lo que puede causar falsas alarmas Zhang *et al.* (2016). Sin embargo, las redes neuronales que se han adoptado para el diagnóstico inteligente de fallas tienen estructuras superficiales Jia *et al.* (2016). Esto limita la capacidad de las ANN para aprender relaciones no lineales complejas, por lo que es necesario establecer una red de arquitectura profunda para este propósito.

Las redes neuronales profundas (RNP) contienen múltiples capas de operaciones no lineales para gestionar estructuras complejas. Estas capas están conectadas entre sí, de modo que los parámetros (pesos) se ajustan a representaciones abstractas de aplicaciones reales. Las RNP capturan funciones complejas mediante el entrenamiento de los múltiples niveles de abstracción, utilizando únicamente los datos directamente del sensor, ya que la red neuronal puede aprender las características del sensor tanto en condiciones de fallo como en estado operativo.

En este trabajo, se propone el uso de cuatro redes neuronales profundas, para la clasificación rápida y precisa de fallas en sensores y actuadores de un motor de inducción, por medio del procesamiento de datos en tiempo real Jia *et al.* (2016). Las redes propuestas son: MLP, LSTM, BiLSTM y CNN.

Los requisitos actuales de tiempo real para el procesamiento de datos en línea, además de la variación de los patrones de comportamiento en los datos observados por el sensor a lo largo del tiempo, reducen la eficacia de los métodos convencionales de aprendizaje automático. Asimismo, las incertidumbres, los elementos no lineales y la variabilidad de la naturaleza de los datos del sensor, que constituyen información valiosa en los procesos de monitoreo para capturar la relación entre datos de series tempo-

rales, requieren un enfoque novedoso para la clasificación de fallas. Estos enfoques deben ser computacionalmente eficientes y tener altos niveles de precisión y exactitud en las predicciones. De este modo, es posible modelar el sistema con fallas en sensores y actuadores, de la siguiente forma:

$$\bar{x}(t+1) = F(\bar{x}(t), \bar{u}(t)) + d(t) \quad (1)$$

$$y(t) = C\bar{x}(t) \quad (2)$$

donde $x \in \mathbb{R}^n$ es el vector de estado del sistema, $u \in \mathbb{R}^m$, $F \in \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$ es una función no lineal, $C \in \mathbb{R}^{p \times m}$ es la matriz de salida y $d \in \mathbb{R}^n$ es el vector de perturbación. Es posible reescribir (1) a partir de sus componentes, como:

$$\bar{x}_i(t+1) = F_i(\bar{x}(t), \bar{u}(t)) + d_i(t) \quad i = 1, 2, \dots, n \quad (3)$$

Las fallas en sensores y actuadores, se definen como:

$$\bar{x}(t) = \rho_i(x_i(t), \delta_i(t)) \quad (4)$$

$$\bar{u}(t) = \phi_i(u_i(t), \delta_i(t)) \quad (5)$$

donde δ_i es el vector de incertidumbres; ρ_i y ϕ_i son funciones no lineales consideradas desconocidas pero acotadas y representan la pérdida de la eficacia en sensores y actuadores causada por entradas externas, no medibles, sesgadas o imprecisas durante un tiempo t_i^* . Entonces, $x_i(t)$ se considera medible y su medida es definida como $\bar{x}_i(t)$. Una representación esquemática de (1) se puede ver en Fig. 1.

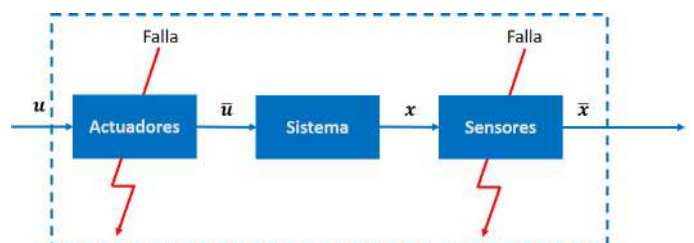


Fig. 1. Representación esquemática de las fallas en sensores y actuadores

Para propósitos de DAF, los valores de \bar{x} y \bar{u} representan la dinámica interna del sistema (1) lo que puede resultar en dos situaciones posibles: falla o no falla. Esto puede considerarse como un problema de clasificación descrito por

$$\rho_i(x_i(t), \delta_i(t)) = \begin{cases} falla, & \rho_i(*) \in S \\ nofalla, & \rho_i(*) \notin S \end{cases} \quad (6)$$

Similarmente,

$$\phi_i(u_i(t), \delta_i(t)) = \begin{cases} falla, & \phi_i(*) \in S \\ nofalla, & \phi_i(*) \notin S \end{cases} \quad (7)$$

donde S es el subconjunto de todos los modos de falla posibles. Por lo tanto, el subconjunto S se considera solo parcialmente conocido debido a la dificultad de definir completamente el sistema no lineal.

Si el estado completo $x(t)$ del sistema está disponible y existe una estructura de datos con la dinámica de (6) y (7) que puede identificarse como características, en una serie de tiempo, se cumplen, entonces podemos afirmar

que es posible identificar un escenario de falla, en el sistema combinando todas las entradas de señal en una serie temporal multivariable.

Para este trabajo, se consideran las redes profundas mencionadas anteriormente para realizar la tarea de clasificación (6) y (7), por lo que los problemas de detección y aislamiento de fallas se dividen en dos etapas: la etapa de aislamiento de la señal y la etapa de clasificación de fallas. La metodología se describe en la siguiente sección.

3. LÓGICA PARA EL AISLAMIENTO DE FALLAS

El conjunto de datos obtenido por el actuador y los sensores puede considerarse una serie de tiempo que relaciona los datos actuales con los pasados, lo que permite corroborar tendencias. Podemos definir una serie temporal como un vector X compuesto por valores reales medidos por sensores propensos a fallos $\bar{x}(k)$, se define como:

$$X = [\bar{x}(0), \bar{x}(1), \dots, \bar{x}(n)] \quad (8)$$

El tamaño del vector corresponde al número de muestras observadas, n . Por lo tanto, existen diferentes maneras de abordar el problema de la DAF; por un lado, utilizando técnicas de clasificación de series temporales multivariables (datos obtenidos a partir de mediciones de múltiples sensores o variables) y por otro, utilizando series temporales univariadas (datos obtenidos a partir de mediciones de la misma variable).

Por lo tanto, N series de tiempo univariadas diferentes $X(k)$, como la ecuación (8), pueden verse como series temporales multivariables que consisten en $X = [X(1), X(2), \dots, X(N)]$. Entonces, el conjunto de datos $D = \{(X(1), Y(1)), (X(2), Y(2)), \dots, (X(N), Y(N))\}$ es una colección donde $X(k)$ es una serie temporal univariada e $Y(k)$ es su vector de etiquetas de clase. La longitud de $Y(k)$ corresponde al número de clases i , donde cada elemento $j \in [1, k]$ es igual a 1 si la clase de $X(k)$ es j y 0 en caso contrario.

Una serie de tiempo univariada presenta el problema de incorporar información, ya que solo se considera disponible una señal. En el caso de las redes neuronales profundas, el contexto y la información proporcionada son cruciales. Por lo tanto, la incorporación del contexto para redes neuronales en series temporales univariadas se puede realizar de dos maneras: mediante conexiones recursivas para modelar el flujo temporal directamente o recopilando entradas en ventanas temporales superpuestas.

La extracción de información de observaciones pasadas se conoce como ventana deslizante o incrustación con retardo temporal Perea and Harer (2015). En este trabajo se propone la ventana deslizante implementada en línea. Para ello, se considera X como una función definida en un intervalo de la serie temporal observada $\{\bar{x}(0), \bar{x}(1), \dots, \bar{x}(t)\}$. Por lo tanto, la incrustación de la ventana deslizante $X(t)$ es un vector de retardo generado mediante la extracción de información local de la serie temporal definida hasta el tiempo actual t .

$$X(t) = [\bar{x}(t - (d - 1)), \dots, \bar{x}(t - 1), \bar{x}(t)] \quad (9)$$

donde $\{1, 2, \dots, d - 1\}$ es el vector de regresión y d es la dimensión del retardo. El vector $X(t)$ es la entrada al

modelo neuronal para predecir la clase $Y(t)$. Es necesario probar diferentes dimensiones del vector de regresión para encontrar el número apropiado de dimensiones que reduce el error de clasificación según la complejidad de la señal.

4. METODOLOGÍA PROPUESTA

En esta propuesta se considera una red neuronal conformada a su vez por cinco clasificadores locales (uno para cada canal o variable a diagnosticar), para el proceso de la detección de fallas y dos neuronas a la salida para la clasificación de las fallas (una neurona de salida para las fallas en sensores y otra neurona de salida para fallas en actuadores), este esquema se puede ver en Fig. 2. Considerando las siguientes etiquetas para las fallas tanto en sensores como en actuadores.

Tabla 1. Etiquetas para la clasificación

Salida 1	Salida 2	Falla
0	0	No falla
1	0	Perturbación o incertidumbre
2	0	Posición
3	0	Corriente i_α
4	0	Corriente i_β
5	0	Posición y corriente i_α
6	0	Posición y corriente i_β
7	0	Corriente i_α y corriente i_β
0	0	No falla
0	1	Voltaje u_α
0	2	Voltaje u_β
0	3	Voltaje u_α y voltaje u_β

4.1 Topología de la red neuronal

Cada uno de los cinco clasificadores se probó con las cinco estructuras neuronales ya mencionadas, considerando las siguientes estructuras

Tabla 2. Estructuras neuronales para cada tipo de red

Red	Arquitectura
MLP	Dos capas ocultas con 20 neuronas cada una
LSTM	Una capa oculta con 15 celdas LSTM
BiLSTM	Una capa con 15 celdas LSTM de estado directo y 15 en la capa oculta de estado inverso
CNN	Una capa de convolución + ReLu con 20 filtros una capa de agrupamiento y dos capas densas

Cada red neuronal tiene como entrada un vector de regresión con diferente dimensión de acuerdo a la variable considerada, de la siguiente manera:

Tabla 3. Dimensión del retardo para cada variable

Variable	Retardo máximo
Posición	$d = 2$
Corriente i_α e i_β	$d = 10$
Voltaje u_α y u_β	$d = 10$

5. RESULTADOS

El entrenamiento se realizó fuera de línea con datos obtenidos experimentalmente de tal forma que los datos utilizados en el entrenamiento sean diferentes a los datos

de prueba. Cabe destacar que las ventanas deslizantes se generaron en diferentes dimensiones para encontrar el número más adecuado de dimensiones en cuanto al error de clasificación. Por su parte, la implementación en línea y en tiempo real utilizó los parámetros ajustados en el entrenamiento de las redes neuronales.

Los resultados se obtuvieron utilizando las cuatro redes neuronales con las estructuras mencionadas anteriormente. Los clasificadores neuronales se entrenaron individualmente para cada canal; se utilizaron 20,000 muestras, como se muestra en Fig. 3. El 50% de los datos utilizados presentaba fallas y el resto no.

Para comparar el rendimiento de los diferentes clasificadores neuronales, utilizamos la exactitud de clasificación, así como el área bajo la curva característica operativa (ROC, por sus siglas en inglés). La ecuación de exactitud de clasificación indica la relación entre el número de predicciones correctas y el número total de muestras, que se obtiene mediante

$$Exactitud = \frac{Positivos\ verdaderos}{Positivos\ verdaderos + Positivos\ falsos} \tag{10}$$

Tabla 4. Resultados obtenidos para la MLP

Sensor	Retardo	ROC	Exactitud
Posición	$d = 2$	1	1
Corriente i_α	$d = 8$	0.9790	0.9665
	$d = 10$	0.9783	0.9666
Corriente i_β	$d = 8$	0.9867	0.9741
	$d = 10$	0.9862	0.9718
Voltaje u_α	$d = 8$	0.9801	0.9678
	$d = 10$	0.9758	0.9759
Voltaje u_β	$d = 8$	0.9883	0.8737
	$d = 10$	0.9846	0.8823

5.1 Análisis comparativo

Los resultados del estudio indican que todas las redes neuronales evaluadas son eficaces en la clasificación de la

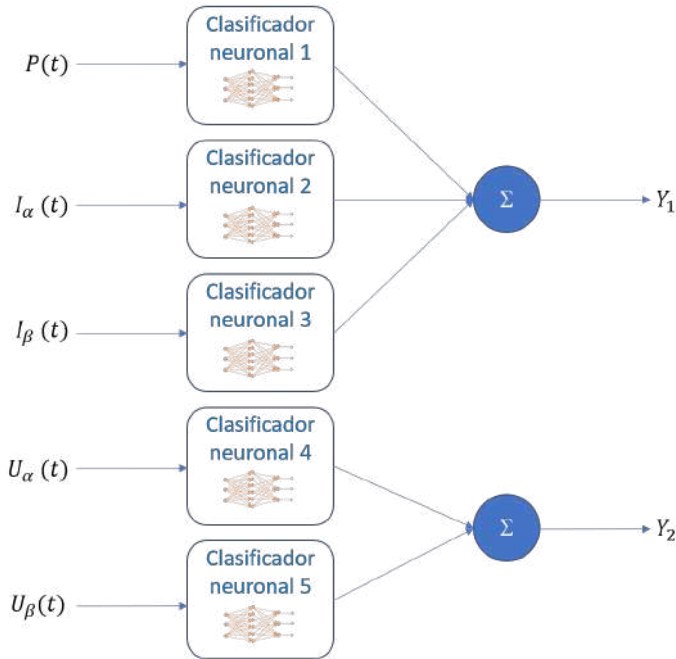


Fig. 2. Estructura neuronal propuesta

falla, destacando la red CNN por su rendimiento superior, con una exactitud que supera al 99 %. A pesar de que las redes MLP, LSTM y BiLSTM también muestran un buen rendimiento, los canales de corrientes (Canal 2 y Canal 3) son los más difíciles de clasificar, presentando bajos

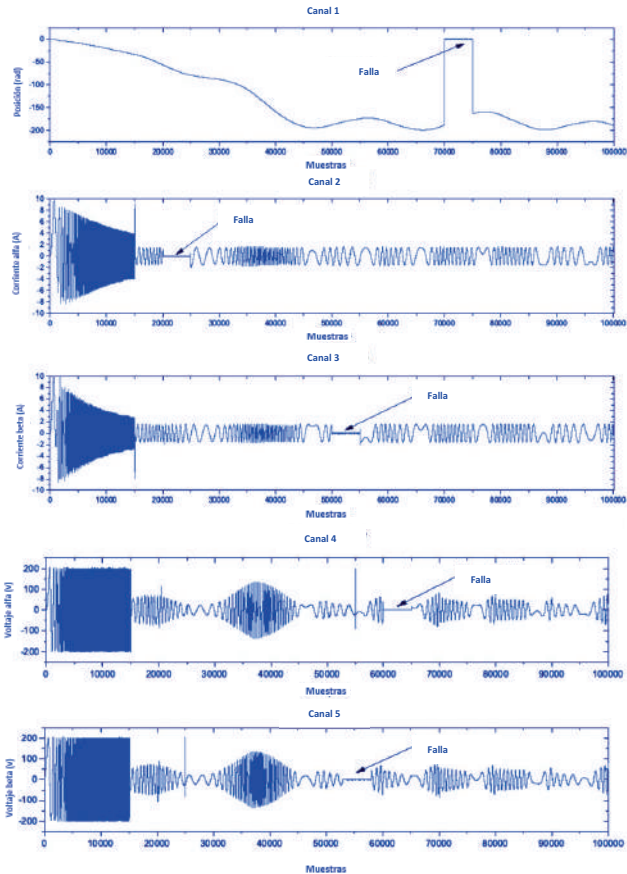


Fig. 3. Serie de tiempo multivariable del motor de inducción

Tabla 5. Resultados obtenidos para la LSTM

Sensor	Retardo	ROC	Exactitud
Posición	$d = 2$	1	1
Corriente i_α	$d = 8$	0.9864	0.9778
	$d = 10$	0.9791	0.9675
Corriente i_β	$d = 8$	0.9954	0.9892
	$d = 10$	0.9961	0.9896
Voltaje u_α	$d = 8$	0.9971	0.9905
	$d = 10$	0.9972	0.9907
Voltaje u_β	$d = 8$	0.9973	0.9911
	$d = 10$	0.9972	0.9845

Tabla 6. Resultados obtenidos para la BiLSTM

Sensor	Retardo	ROC	Exactitud
Posición	$d = 2$	1	1
Corriente i_α	$d = 8$	0.9873	0.9381
	$d = 10$	0.9787	0.9318
Corriente i_β	$d = 8$	0.9914	0.9823
	$d = 10$	0.9902	0.9821
Voltaje u_α	$d = 8$	0.9962	0.9887
	$d = 10$	0.9964	0.9890
Voltaje u_β	$d = 8$	0.9971	0.9907
	$d = 10$	0.9965	0.9909

rendimientos. En contraste, los canales actuadores (Canal 4 y Canal 5) mostraron un desempeño adecuado, especialmente en las redes LSTM, BiLSTM y CNN, alcanzando igualmente una exactitud superior al 99%.

La red neuronal MLP destaca por su tiempo promedio de ejecución por muestra, siendo la mejor opción para experimentos en tiempo real, aunque con un ligero sacrificio de precisión en comparación con CNN y LSTM, que ofrecen mejores resultados. Sin embargo, si la velocidad de clasificación por muestra no es una prioridad, el modelo CNN se considera la mejor alternativa.

6. CONCLUSIONES

La clasificación de series de tiempo es relevante en diversas aplicaciones, como en el ámbito médico e industrial. Los métodos tradicionales suelen ser costosos y complejos, lo que limita su uso. Este trabajo aborda la detección y aislamiento de fallas en sensores y actuadores de un motor de inducción mediante clasificadores neuronales en tiempo real, sin requerir un modelo matemático del sistema ni redundancia de sensores. Se implementó un enfoque sencillo donde se consideran diferentes señales como canales, permitiendo que los clasificadores aprendan a clasificar series temporales univariantes. Además, se integró contexto para el entrenamiento de las redes neuronales a través de un vector de regresión.

Una posible modificación para mejorar el desempeño de la clasificación consistiría en emplear una combinación de redes neuronales, específicamente una red MLP para el canal 1 y una red LSTM para los canales 2 y 3. Los resultados obtenidos son prometedores, dado que se basa únicamente en la información de los tres sensores disponibles: posición, corriente.

7. AGRADECIMIENTOS

Los autores agradecen el apoyo del CUCEI, de la Universidad de Guadalajara por el apoyo proporcionado para el desarrollo de esta investigación.

REFERENCIAS

Sanchez, Oscar D. and Martinez-Soltero, Gabriel and Alvarez, Jesus G. and Alanis, Alma Y. Real-Time Neural Classifiers for Sensor and Actuator Faults in Three-Phase Induction Motors. *Machines* **2022**, *10*, 1198.

Rodriguez-Guerra, J.; Calleja, C.; Pujana, A.; Elorza, I.; Macarulla, A.M. Fault-tolerant control study and classification: Case study of a hydraulic-press model simulated in real-time. *Int. J. Electr. Inf. Eng.* **2019**, *13*, 115–127.

Tabla 7. Resultados obtenidos para la CNN

Sensor	Retardo	ROC	Exactitud
Posición	$d = 2$	1	1
Corriente i_α	$d = 8$	0.9845	0.9665
	$d = 10$	0.9965	0.9909
Corriente i_β	$d = 8$	0.9878	0.9731
	$d = 10$	0.9983	0.9942
Voltaje u_α	$d = 8$	0.9963	0.9807
	$d = 10$	0.9987	0.9957
Voltaje u_β	$d = 8$	0.9920	0.9753
	$d = 10$	0.9985	0.9930

Chen, H.; Jiang, B.; Ding, S.X.; Huang, B. Data-driven fault diagnosis for traction systems in high-speed trains: A survey, challenges, and perspectives. *IEEE Trans. Intell. Transp. Syst.* **2020**, *23*, 1700–1716.

Saufi, S.R.; Ahmad, Z.A.B.; Leong, M.S.; Lim, M.H. Challenges and opportunities of deep learning models for machinery fault detection and diagnosis: A review. *IEEE Access* **2019**, *7*, 122644–122662.

Zhang, J.; Swain, A.K.; Nguang, S.K. *Robust Observer-Based Fault Diagnosis for Nonlinear Systems Using MATLAB®*; Springer: Berlin/Heidelberg, Germany, 2016.

Jia, F.; Lei, Y.; Lin, J.; Zhou, X.; Lu, N. Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data. *Mech. Syst. Signal Process.* **2016**, *72*, 303–315.

Perea, J.A.; Harer, J. Sliding windows and persistence: An application of topological methods to signal analysis. *Found. Comput. Math.* **2015**, *15*, 799–838.