RIELAC
Revista de Ingeniería Electrónica, Automática y Comunicaciones

Sistema de Monitoreo basado en Aprendizaje Profundo en Sistemas Industriales

Adrián Rodríguez Ramos, Orestes Llanes-Santiago

RESUMEN / *ABSTRACT*

El paradigma Industria 4.0 tiene como objetivo obtener altos niveles de productividad y eficiencia, productos finales más competitivos y el cumplimiento de las exigentes normativas relacionadas con la seguridad industrial y la ciberseguridad. Para lograr estos objetivos, los sistemas industriales deben estar equipados con sistemas de monitoreo de condición para la detección temprana y localización de fallos y ciberataques. Este artículo propone una estrategia robusta de monitoreo de condición mediante el uso de algoritmos de Aprendizaje Profundo. El esquema propuesto fue validado utilizando un proceso de prueba Tennessee Eastman (TE) con excelentes resultados. La estrategia propuesta se comparó con otros esquemas de monitoreo de condición. El mayor rendimiento obtenido por el esquema propuesto indica su factibilidad.

Palabras claves: Industria 4.0, Ciberseguridad, Diagnóstico de fallos, Aprendizaje profundo, sistema de monitoreo

The Industry 4.0 paradigm aims to obtain high levels of productivity and efficiency, more competitive final products, and compliance with demanding regulations related to industrial safety and cyber-security. To achieve these goals, industrial systems must be equipped with condition monitoring systems for the early detection and localization of faults and cyber-attacks. This paper proposes a robust condition-monitoring strategy through the use of Deep Learning algorithms. The proposed scheme was validated using a Tennessee Eastman (TE) process with excellent results. The proposed strategy was compared with other condition monitoring schemes. The higher performance obtained by the proposed scheme indicates its feasibility.

Keywords: Industry 4.0, Cybersecurity, Fault diagnosis, Deep learning, monitoring system.

Monitoring system based on deep learning in industrial systems

1. -INTRODUCCIÓN

Términos como Fábrica Inteligente e Industria 4.0 son conceptos emergentes que describen la aplicación en el entorno industrial y empresarial de un amplio conjunto de tecnologías relacionadas con la digitalización, la conectividad y la automatización [1]. Estas tecnologías incluyen el Internet de las Cosas (*IoT*), la Computación en la Nube (*Cloud Computing*), Manejo de Grandes Cantidades de Datos (*Big Data*), Inteligencia Artificial (IA), la Robótica (R), las Cadenas de Bloques (*Blockchain*) y la Ciberseguridad (*Cybersecurity)* y se caracterizan por sistemas cada vez más conectados y mayor integración de las tecnologías digitales en diferentes procesos [2].

Las practicas industriales y empresariales evolucionan en paralelo a esta innovación tecnológica y están surgiendo nuevos modelos de negocio con productos inteligentes que exigen un cambio hacia los servicios digitales. Este nivel superior de integración permite niveles más altos de productividad, productos finales más competitivos, y excelente cumplimiento de las normas de seguridad industrial.

Recibido: 11/2022 Aceptado: 03/2023

Sin embargo, para lograr estos importantes beneficios comerciales, dos cuestiones cruciales aumentan su riesgo y deben ser atendidas con prioridad: la ciberseguridad industrial [3-5] y la ocurrencia de fallos debido a sus efectos adversos sobre la seguridad industrial, la productividad y los gastos operativos [6-8].

Para hacer frente a estos problemas, las plantas industriales deben estar equipado con sistemas de monitoreo de condición para la detección temprana y localización de ataques cibernéticos y fallos. Esto ha guiado una gran cantidad de investigaciones sobre estos temas en los últimos años sin embargo se abordan de forma independiente [9-11]. Por otra parte, para lograr el mejor rendimiento en el proceso de diagnóstico, las clases que representan los modos de operación de la planta industrial deben estar muy bien identificadas [12]. Sin embargo, este es un tema muy complejo debido a las incertidumbres que caracterizan las mediciones industriales por el efecto del ruido y las perturbaciones.

En la literatura científica, la detección y localización de ciberataques y los esquemas de diagnóstico de fallos se clasifican en: métodos basados en modelos y basados en datos [13, 14]. El uso exitoso de los métodos basados en modelos de proceso depende de una comprensión profunda del proceso, su funcionamiento, parámetros y modos de funcionamiento. Sin embargo, la gran complejidad de las plantas industriales actuales dificulta el logro de este conocimiento. El segundo grupo incluye los métodos basados en datos, que no necesitan un modelo matemático preciso o un conocimiento profundo inicial de los parámetros del proceso y la correlación entre variables [15-18]. Los avances en las tecnologías de Internet de las Cosas (*IoT*) y el *Big Data* han permitido actualmente una mayor atención y resultados en este último enfoque [19-21].

Una revisión de las estrategias de monitoreo de condición en los últimos años muestra un aumento considerable en el uso de técnicas de Inteligencia Artificial (IA). Las técnicas de IA pueden identificar fácilmente el problema presente en la aplicación y pueden tomar medidas sólidas. También se utiliza para procesar la gran cantidad de información que los usuarios generan a diario. El aprendizaje profundo es una de las poderosas técnicas de aprendizaje automático impulsadas por la IA. Las técnicas de aprendizaje profundo pueden procesar una gran cantidad de información presente en los conjuntos de datos de los sistemas de monitoreo de manera eficiente [22]. Estas técnicas tienen en cuenta las incertidumbres e imprecisiones presentes en los datos durante el entrenamiento y la validación para superar la superposición entre las clases de fallos y ataques. Por lo tanto, esto permite mejorar la eficiencia y la robustez contra incertidumbres, lo que proporciona una mejor precisión en la clasificación de fallos y ataques en comparación con otras técnicas existentes.

La revisión de la literatura científica actual también mostró que la detección y localización de fallos y ciberataques se han abordado por separado a pesar de ser objetivos del sistema de monitoreo de condición [11,20,23-25]. En este trabajo se propone una nueva estrategia de monitorización de condiciones mediante técnicas de aprendizaje profundo que aborda integralmente el diagnóstico de fallos y ciberataques en plantas industriales. La principal contribución de este trabajo es presentar una estrategia robusta de monitoreo de condición frente a perturbaciones externas y ruido, con la capacidad de detectar y localizar la ocurrencia de fallos y ataques cibernéticos. Para ello, se presenta un esquema basado en el uso de algoritmos de aprendizaje profundo. Los experimentos desarrollados mostraron un alto rendimiento de la estrategia propuesta en presencia de ruido.

La estructura del documento es la siguiente: la Sección 2 sobre Materiales y Métodos describe las características generales sobre las técnicas de aprendizaje profundo, en particular Redes Neuronales Recurrentes (*RNN*). Además, se presenta el esquema de monitoreo propuesto utilizando una red *Long Short-Term Memory* (*LSTM*) y se realiza la descripción del proceso Tennessee Eastman (TE) usado para validar los experimentos realizados. En la Sección 3 se analizan y discuten los resultados obtenidos. Por último, se exponen las conclusiones.

2.- Materiales y Métodos

2.1.- Características generales sobre las técnicas de aprendizaje profundo

Los enfoques de aprendizaje profundo se pueden clasificar de la siguiente manera: supervisados, semisupervisados o parcialmente supervisado y no supervisado. Además, hay otra categoría de enfoque de aprendizaje llamados Aprendizaje por Refuerzo (*RL*) o RL Profundo (*DRL*) que a menudo se discuten bajo el alcance de enfoques de aprendizaje semisupervisados o, a veces, no supervisados [22,25].

El aprendizaje supervisado es una técnica de aprendizaje que utiliza datos etiquetados. En el caso de los enfoques de aprendizaje profundo supervisados, el entorno tiene un conjunto de entradas y salidas correspondientes $(x_t, y_t) \sim \rho$. Por ejemplo, si para la entrada x_t, el agente inteligente predice $\hat{y}_t = f(x_t)$, el agente recibirá un valor de pérdida $l(y_t, \hat{y}_t)$. A continuación, el agente modificará iterativamente los parámetros de red para una mejor aproximación de las salidas deseadas. Después de un entrenamiento exitoso, el agente podrá obtener las respuestas correctas a las preguntas del entorno. Existen diferentes enfoques de aprendizaje supervisado para el aprendizaje profundo, incluidas las Redes Neuronales Profundas (*DNN*), las Redes Neuronales Convolucionales (*CNN*), Redes Neuronales Recurrentes (*RNN*), incluida la de Memoria a Largo Corto Plazo (*LSTM*) y las Unidades Recurrentes Cerradas (*GRU*).

El aprendizaje semisupervisado es el aprendizaje que se produce en función de conjuntos de datos parcialmente etiquetados. En algunos casos, DRL y *Generative Adversarial Networks* (*GAN*) se utilizan como técnicas de aprendizaje semisupervisado. Además, las *RNN*, también se utilizan para el aprendizaje semisupervisado.

Los sistemas de aprendizaje no supervisado son aquellos que funcionan sin la presencia de etiquetas de datos. En este caso, el agente aprende la representación interna o las características importantes para descubrir relaciones o estructuras desconocidas dentro de los datos de entrada. A menudo, la agrupación, la reducción de la dimensionalidad y las técnicas generativas se consideran enfoques de aprendizaje no supervisado.

2.1.1.- Redes Neuronales Recurrentes (RNN)

Normalmente el aprendizaje de algún conocimiento nuevo está basado en la comprensión y aprendizaje de conocimientos anteriores. Los enfoques tradicionales de redes neuronales, incluidas las *DNN* y las *CNN*, no pueden lidiar con este tipo de problema. Las redes neuronales estándar y *CNN* son incapaces de lograrlo debido a las siguientes razones. Primero, estos enfoques solo manejan un vector de tamaño fijo como entrada (por ejemplo, una imagen o un cuadro de video) y producen un vector de tamaño fijo como salida (por ejemplo, probabilidades de diferentes clases). En segundo lugar, esos modelos operan con un número fijo de pasos computacionales (por ejemplo, el número de capas en el modelo). Las *RNN* son únicas, ya que permiten la operación sobre una secuencia de vectores a lo largo del tiempo. *Long Short-Term Memory* (LSTM) es uno de los modelos de aprendizaje profundo más efectivos en el procesamiento secuencial de datos. Es considerada la forma especializada de la Red Neuronal Recurrente (*RNN*) que puede almacenar información en una célula de memoria durante un período más prolongado [6,8].

Una celda *LSTM* tiene tres puertas: olvido, entrada y salida que controlan y protegen los estados de la celda. Las puertas controlan el acceso y el flujo de información en la celda de memoria y evitan que la información almacenada sea sobrescrita con información irrelevante. Al igual que *RNN, LSTM* tiene una capa de entrada que se muestra como $\boldsymbol{x_t}$, y la salida se muestra como $\boldsymbol{h_t}$ en el tiempo ***t***. *LSTM* toma el estado de entrada de la celda actual $\widetilde{\boldsymbol{C_t}}$, estado de salida de celdas anteriores $\boldsymbol{C_{t-1}}$ y genera un estado de salida de $\boldsymbol{C_t}$. La Figura 1 muestra el diagrama de arquitectura de una celda *LSTM*, donde puede apreciarse las tres puertas como: puerta de entrada ($\boldsymbol{i_t}$), puerta de olvido ($\boldsymbol{f_t}$) y salida ($\boldsymbol{o_t}$), respectivamente [23].

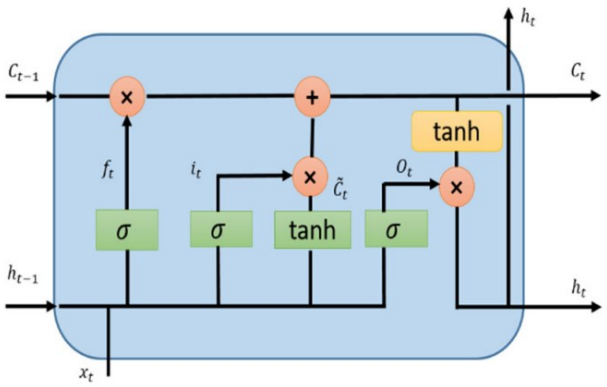

Figura 1

Diagrama para LSTM [22]

Las ecuaciones 1-4 representan la operación de LSTM, donde $\boldsymbol{f_i^{(t)}}$ en la ecuación (1) representa la puerta de olvido en el paso de tiempo ***t*** en la celda ***i.*** Se aplica una función sigmoide σ para establecer el valor de peso entre 0 y 1. En el paso de tiempo

t, ***x(t)*** representa el vector de entrada actual y ***h(t)*** representa el vector en la capa oculta. Los *bias* $\boldsymbol{b^f}$, $\boldsymbol{b^g}$, $\boldsymbol{b^o}$ se agregan a todas las células *LSTM*. Los pesos de entrada $\boldsymbol{U^f}$, $\boldsymbol{U^g}$, $\boldsymbol{U^o}$ se agregan a las entradas actuales y a los pesos recurrentes $\boldsymbol{W^f}$, $\boldsymbol{W^g}$, $\boldsymbol{W^o}$ se agregan a los vectores de la capa oculta. La ecuación (2) muestra el estado interno de una celda *LSTM*, y la ecuación (3) presenta la unidad de puerta de entrada externa $\boldsymbol{g_i^{(t)}}$. La ecuación (4) representa la puerta de salida $\boldsymbol{q_i^{(t)}}$ en el paso de tiempo ***t***

$$f_i^{(t)} = \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}\right) \quad (1)$$

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma\left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)}\right) \quad (2)$$

$$g_i^{(t)} = \sigma\left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}\right) \quad (3)$$

$$q_i^{(t)} = \sigma\left(b_i^o + \sum_j U_{i,j}^o x_j^{(t)} + \sum_j W_{i,j}^o h_j^{(t-1)}\right) \quad (4)$$

2.2.- Metodología propuesta

Para entrenar una red neuronal que puede detectar y localizar los fallos y ciberataques en un proceso industrial debe seguirse el siguiente flujo de trabajo: i) Preprocesar los datos ii) Diseñar la arquitectura de la red iii) Entrenar la red y iv) Realizar validación. La Figura 2 muestra el esquema general de monitoreo propuesto para detectar y localizar la ocurrencia de fallos y ataques cibernéticos en un sistema industrial.

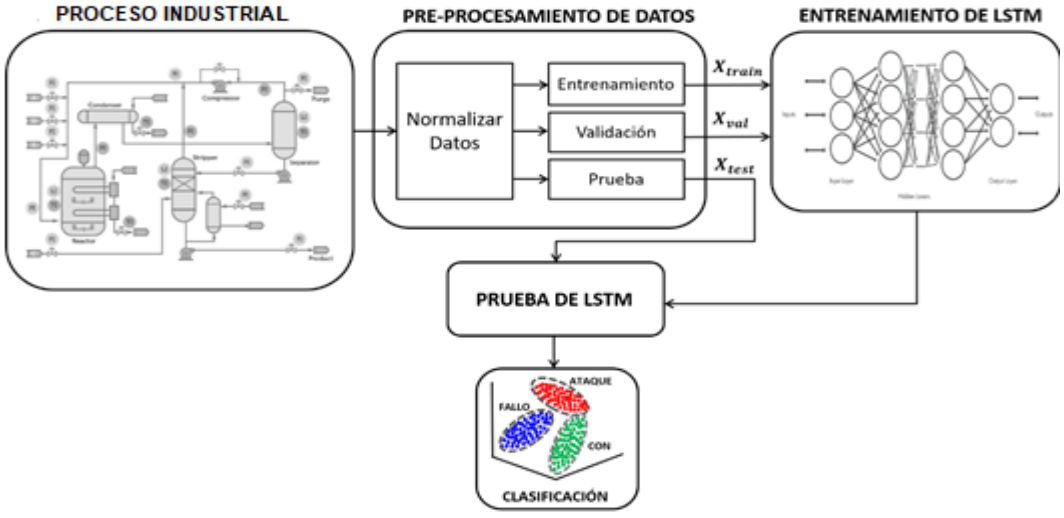

Figura 2
Esquema de monitoreo propuesto en el problema de prueba Tennessee Eastman (TE)

En la etapa de preprocesamiento lo primero es normalizar los datos para asegurar que una variable con un valor mayor no domine otras variables sin perder ninguna información importante requerida para el entrenamiento. Posteriormente, se dividen

los datos en datos de entrenamiento y validación dedicando para esta última acción el 20 por ciento del total de los datos. El uso de un conjunto de datos de validación permite evaluar el ajuste del modelo con el conjunto de datos de entrenamiento mientras se ajustan los parámetros del modelo. En el entrenamiento hay que evitar que la red se sobreajuste y pierda capacidad de generalización. Finalizado el entrenamiento, se ejecuta la validación y se predice el estado en el que se encuentra el proceso, calculando la precisión (*accuracy*) para cada clase.

2.3.- Caso de Estudio: Tennessee Eastman (TE)

2.3.1.- Descripción del proceso

El proceso Tennessee Eastman (TE) se utiliza para evaluar el rendimiento del nuevo esquema de monitoreo [11]. La planta consta de cinco subprocesos interconectados entre sí (ver Figura 3). En el proceso se obtienen los productos, G y H, de los reactivos A, C, D y E. La planta se divide en cinco componentes tecnológicos: un reactor, un condensador, un separador vapor-líquido, un decapante de productos y un compresor.

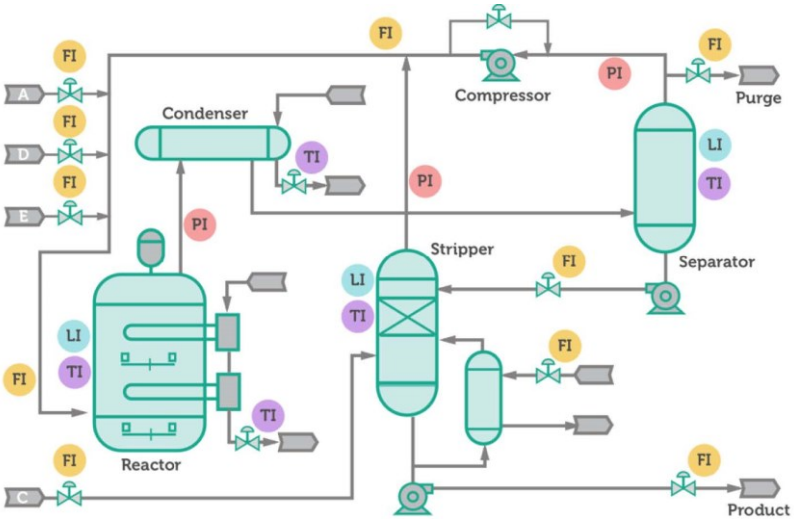

Figura 3
Diagrama de tuberias del proceso TE

Este proceso tiene 52 variables y un conjunto de observaciones correspondientes a la Condición de Operación Normal (CON) y 21 fallos. Cada conjunto de observación se genera a lo largo de 48 horas y los fallos se incorporan después de 8 horas de simulación, incorporando ruido en las mediciones con lo cual se puede probar la robustez del sistema propuesto a la presencia de ruido y perturbaciones (Obtenido de: http://web.mit.edu/braatzgroup/TE_proceso.zip). La Tabla 1 muestra los fallos utilizados para el análisis y evaluación del esquema propuesto. Para la etapa de entrenamiento (*Training*), se utilizaron 500 muestras de cada fallo y la CON, y se consideraron 960 muestras en la etapa de prueba (*Testing*).

Tabla 1
Fallos considerados en el proceso TE

Fallo	Variables del proceso	Tipo
F1	A/C relación de alimentación, B constante de composición	Paso
F6	A pérdida de alimentación	Paso
F7	C disponibilidad reducida de pérdida de presión del cabezal	Paso

La Figura 4 muestra el comportamiento de algunas variables síntomas en presencia del fallo F6. Aunque la ocurrencia del fallo puede causar problemas en el funcionamiento del proceso, en ningún momento el sistema se apaga. A continuación, se analizará qué sucede cuando el proceso está bajo ciberataques. Los ataques simulados se muestran en la Tabla 2, donde el

atacante tiene pleno conocimiento del proceso y es capaz de falsificar una medición del sensor en cualquier momento. En los experimentos realizados, se utilizó el Simulink/MATLAB. Al igual que en los fallos simulados, se consideraron 500 observaciones para cada ataque durante el entrenamiento y 960 para la etapa de prueba.

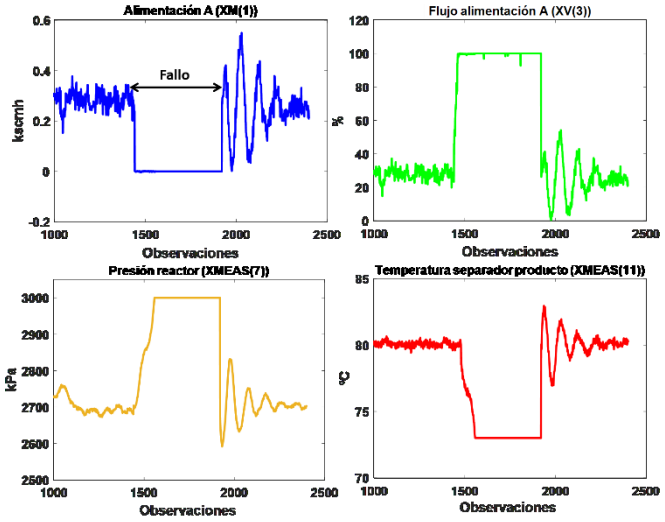

Figura 4
Ocurrencia del fallo 6 en el proceso TE

Tabla2
Descripción de los ataques en el proceso TE

Ataque	Sensor atacado [magnitud]	Variables síntomas	Descripción	Impacto
A1	XM(1) [+2.35]	XM(1), XM(7), XM(8), XV(3)	Durante 3h, el valor real es aumentado en 2.35	Alta presión en el reactor o bajo nivel de *stripper*. Apagado
A2	XM(14) [+7]	XM(12), XM(14), XM(15), XV(7), XV(8)	Durante 2.88h, el valor real es aumentado en 7	Alto nivel en el *stripper*. Apagado
A3	XM(14) [-7]	XM(12), XM(14), XM(15), XV(7), XV(8)	Durante 2.02h, el valor real es disminuido en 7	Bajo nivel en el separador. Apagado

XM(1): Alimentación A, XM(7): Presión reactor, XM(8): Nivel reactor, XM(12): Nivel separador, XM(14): Desbordamiento separador, XM(15): Nivel *stripper,* XV(3): Flujo alimentación A, XV(7): Flujo separador, XV(8): Flujo *stripper*

La Figura 5 muestra la ocurrencia del Ataque 2. Cuando se produce el ataque, la lectura en el sensor de desbordamiento del separador aumenta en 7. Esto hace que el controlador de flujo del separador cierre la válvula para contrarrestar el aumento del caudal. Como resultado del cierre de la válvula, el flujo disminuye (aunque el ataque todavía indica valores más altos), lo que hace que el nivel comience a subir.

El aumento de nivel provoca una apertura gradual de la válvula hasta que alcanza un valor estable. En el mismo intervalo de tiempo, el flujo del separador provoca un aumento en el nivel. Cuando finaliza el ataque, el sensor detecta una medición real más baja del desbordamiento del separador en comparación con el valor informado anteriormente. Como resultado, la válvula se abre y permite que, entre un gran flujo desde el separador hacia el *stripper*, lo que hace que el proceso entre en un modo inseguro y, por lo tanto, se apague el sistema. El objetivo principal es la detección y localización temprana del ataque para evitar el apagado del sistema. En este trabajo la arquitectura de la red *LSTM* fue configurada siguiendo los siguientes pasos:

- Definir que la capa de entrada sea del mismo tamaño que el número de señales de entrada (52).
- Definir 3 capas ocultas con 52, 40 y 25 unidades. Esta selección se basa en el experimento realizado en [8].
- Finalmente, para la clasificación, se incluye una capa totalmente conectada del mismo tamaño que el número de clases de salida (7: CON, 3 fallos y 3 ataques).

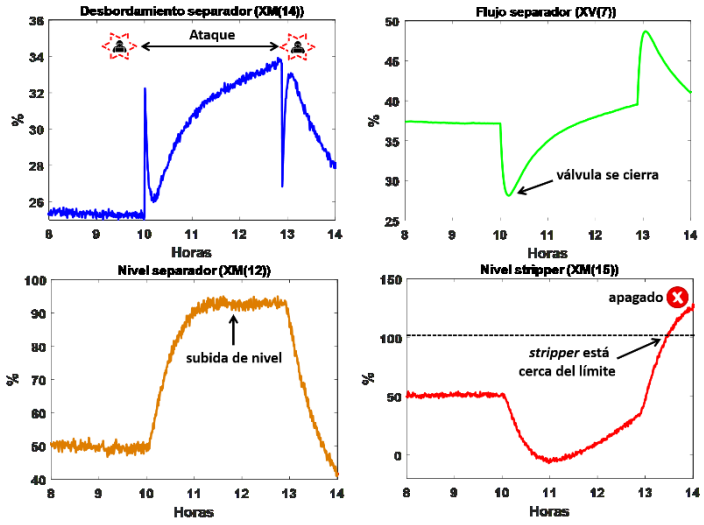

Figura 5
Ocurrencia del ataque A2 en el proceso TE

3.- Análisis y discusión de resultados

Es de gran importancia analizar la calidad y realización de un sistema de monitoreo de condición. Una herramienta ampliamente utilizada para este propósito es la matriz de confusión (MC), que permite el análisis del rendimiento de los algoritmos de clasificación. Los valores MC_{rs} para $r \neq s$ en la MC muestran el número de observaciones del modo de operación r que el algoritmo clasifica erróneamente en los modos de operación. La Tabla 3 muestra la matriz de confusión, donde: CON: Condición de Operación Normal, F1: Fallo 1, F6: Fallo 6, F7: Fallo 7, A1: Ataque 1, A2: Ataque 2 y A3: Ataque 3. La diagonal de la MC corresponde a la cantidad de observaciones que fueron clasificadas con precisión. Conociendo el número total de observaciones en cada clase, puede calcular la precisión en la clasificación (TA = observaciones correctamente clasificadas/observaciones totales). La última fila muestra la precisión promedio de todas las clases.

Como se observa en la Tabla 4, para el caso de los fallos, estos resultados fueron comparados con los obtenidos usando otras redes neuronales: *Hierarchical Neural Network* (HNN), *Shallow Neural Network* (SNN), *Sparse Auto-Encoder* (SAE) y los obtenidos en [8]. Por otra parte, en la Tabla 5 se muestra una comparación con técnicas utilizadas en [25] para la clasificación de los ataques analizados (LR: *Logistic Regression,* LRCV: *Logistic Regression* CV, NN: *Neural Network*, RF: *Random Forest).* Las Figura 6 y 7 muestran los resultados de clasificación (Fallos y Ataques) en el proceso TE, demostrando el alto desempeño del esquema de monitoreo propuesto.

Tabla 3
MC: LSTM (CON: 500, F1: 500, F6: 500, F7: 500, A1: 500, A2: 500, A3: 500)

	CON	F1	F6	F7	A1	A2	A3	TA (%)
CON	**500**	0	0	0	0	0	0	100
F1	0	**500**	0	0	0	0	0	100
F6	0	0	**500**	0	0	0	0	100
F7	0	0	0	**500**	0	0	0	100
A1	0	2	4	0	**494**	0	0	98.80

A2	0	0	0	4	0	**496**	0	99.20
A3	0	0	0	0	0	0	**500**	100
AVE								**99.71**

Tabla 4.
Resultados de la comparación para los fallos (valores en negrita indican el mejor rendimiento)

Fallo	SNN	HNN	SAE	NN [8]	LSTM
1	81.19	97.51	98.75	99.31	**100**
6	83.31	99.38	97.70	**100**	**100**
7	81.49	**100**	99.37	99.86	**100**
AVE	81.99	98.96	98.60	99.72	**100**

Tabla 5.
Resultados de la comparación para los ataques (valores en negrita indican el mejor rendimiento)

Ataque	LR	LRCV	NN	RF	LSTM
1	63.30	63.30	90.10	95.90	**98.80**
2	86.50	86.50	97.00	**100**	99.20
3	76.00	74.00	90.20	93.80	**100**
AVE	75.27	74.60	92.43	96.57	**99.33**

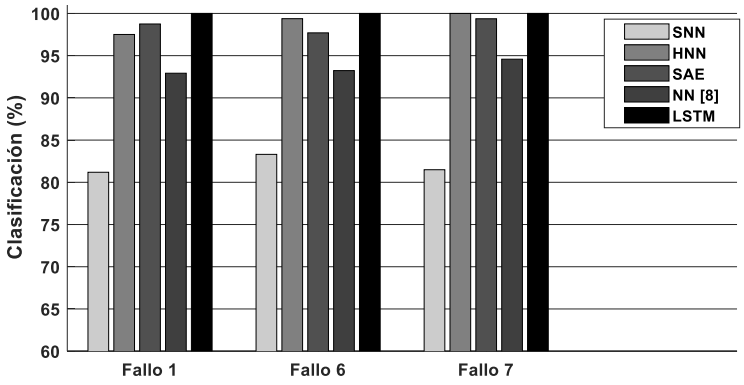

Figura 6.
Clasificación de fallos en el proceso TE.

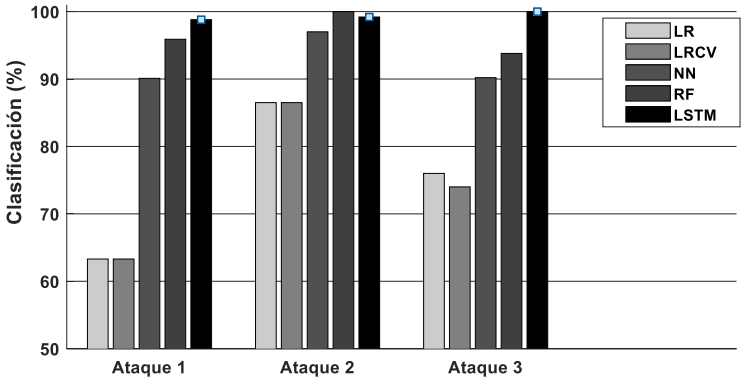

Figura 7.
Clasificación de ataques en el proceso TE.

4.- CONCLUSIONES

Se ha presentado una estrategia robusta para el monitoreo en plantas industriales mediante el uso de algoritmos de aprendizaje profundo, que representa una de las contribuciones científicas del trabajo. El esquema propuesto constituye la principal contribución, el cual integra la detección y clasificación de fallos y ciberataques obteniendo excelentes resultados. El aprendizaje profundo se utiliza con el objetivo de procesar una gran cantidad de información presente en los conjuntos de datos de los sistemas de monitoreo de manera eficiente, teniendo en cuenta las incertidumbres e imprecisiones presentes en los datos para mejorar la clasificación ante la superposición entre las clases de fallos y ataques, mejorando de esta manera la robustez del sistema de monitoreo.

En la propuesta, el entrenamiento fuera de línea es una etapa muy importante. Se utiliza el algoritmo *LSTM* como herramienta de clasificación, para predecir el estado en el que se encuentra el proceso (CON, Fallo y Ataque). El enfoque propuesto fue validado utilizando el proceso de prueba TE, con resultados gratificantes. Finalmente, se hizo una comparación con varios tipos de algoritmos de clasificación. En todos los casos, la estrategia propuesta muestra el mejor rendimiento.

AGRADECIMIENTOS

Los autores agradecen el soporte financiero para el desarrollo de la investigación al Ministerio de Ciencia, Tecnología y Medio Ambiente de Cuba a través del Programa Nacional de Investigación en Automática, Robótica e Inteligencia Artificial (ARIA) y el proyecto PN223LH004-023.

REFERENCIAS

1. Macas M. and Wu C. and Fuertes W., A survey on deep learning for cybersecurity: Progress, challenges, and opportunities, Computer Networks, 2022; 212: 1 – 33.
2. Bashendy M., Tantawy A., Erradi A., Intrusion response systems for cyber-physical systems: A comprehensive survey. Computers & Security, 2023, 124:1 – 27.
3. Azzam M., Pasquale L., Provan G., Nuseibeh B., Forensic readiness of industrial control systems under stealthy attacks, Computers & Security, 2023, 125:1 – 10.

4. Alanazi M. and Mahmood A. and Morshed M.J., SCADA vulnerabilities and attacks: A review of the state of the art and open issues, Computers & Security 2023, 125:1 – 29.

5. Alladi T. and Chamola V. and Zeadally S., Industrial Control Systems: Cyberattack trends and countermeasures, Computer Communications, 2020, 155:1 – 8.

6. Li W. and Huang R. and Li J. and Liao Y. and Chen Z. and He G. and Yan R. and Gryllias. Simons K., A perspective survey on deep transfer learning for fault diagnosis in industrial scenarios: Theories, applications and challenges, Mechanical Systems and Signal Processing 2022, 167:108487.

7. Lv H. and Chen J. and Pan T. and Zhang T. and Feng Y. and Liu S., Attention mechanism in intelligent fault diagnosis of machinery: A review of technique and application, Measurement, 2022, 199:111594.

8. Heo S. and Lee J.H., Fault detection and classification using artificial neural network, IFAC PaperOnLine, 2018, 51(18):470 – 475.

9. Zang P. and Wen G. and Dong S. and Lin H. and Huang X. and Tian X. and Chen X., A novel multiscale lightweight fault diagnosis model based on the idea of adversarial learning, Neurocomputing, 2018, 275:1674 – 1683.

10. Doing D. and Han Q.L. and Xiang Y. and Zhang X.M., The idea of On-line Diagnostics as a Method of Cyberattack Recognition, Advanced Solutions in Diagnostics and Fault Tolerant Control, Springer, 2018, p. 449 – 457.

11. Kravchik M., Demetrio L., Biggio L., Shabtai A., Practical Evaluation of Poisoning Attacks on Online Anomaly Detectors in Industrial Control System, Computers & Security, 2022, 122:1 – 20.

12. Verron S. and Tiplica T. and Kobi A., New informative features for fault diagnosis by supervised classification, 18th Mediterranean Conference on Control and Automation (MED'10), Marrakech, Morocco, 2010, p. 454-459.

13. Wu B. and Cai W. and Chen H. and Zhang X., A hybrid data-driven simultaneous fault diagnosis model for air handling units, Energy and Buildings, 2021, 245:1 – 10.

14. Doing D. and Han Q.L. and Xiang Y. and Zhang X.M., New Features for Fault Diagnosis by Supervised Classification, IEEE Transactions on Instrumentation and Measurement, 2021, 70:1 – 15.

15. Kumar N. and Mohan Mishra and Kumar A., Smart grid and nuclear power plant security by integrating cryptographic hardware chip, Nuclear Engineering and Technology, 202, 53:3327 – 3334.

16. Taqvi S.A.A. and Zabiri H. and Tufa L.D. and Uddinn F. and Fatima S.A. and Maulud A.S., A review on data-driven learning approaches for fault detection and diagnosis in chemical process, ChemBioEng Reviews. 8 (2021) 239 – 259.

17. Hadroug N. and Hafaifa A. and Alili B. and Iratni A. and Chen X., Fuzzy Diagnostic Strategy Implementation for Gas Turbine Vibrations Faults Detection: Towards a Characterization of Symptom Fault Correlations, Journal of Vibration Engineering & Technologies, 2022, 10:225 – 251.

18. Lundgren A. and Jung D., Data-driven fault diagnosis analysis and open-set classification of time-series data, Control Engineering Practice, 2022, 121:105006.

19. Rodríguez-Ramos A. and Bernal-de-Lázaro J.M. and Cruz-Corona C. and Silva Neto A. and Llanes-Santiago O., An approach to robust condition monitoring in industrial processes using pythagorean memberships grades, Annals of the Brazilian Academy of Sciences, 2022, 94:1 – 22.

20. Rodríguez-Ramos A. and Bernal de Lázaro J.M. Prieto-Moreno A., Silva Neto A.J. and Llanes-Santiago O. An approach to robust fault diagnosis in mechanical systems using computational intelligence, Journal of Intelligent Manufacturing, 2019, 30(4):1601-1615.

21. Chi Y. and Dong Y. and Wang Z.Y. and Yu F.R. and Leung V.C.M., Knowledge-Based Fault Diagnosis in Industrial Internet of Things: A Survey, IEEE Internet of Things Journal, 2022, 9:12886 – 12900.

22. Li, Y.; Song, H.; Ly, Z.: Deep learning in security of internet of things. IEEE Internet Things Journal, 2021, 9:22133-22146.

23. Dixit P. and Silakari S., Deep Learning Algorithms for Cybersecurity Applications: A Technological and Status Review, Computer Science Review, 2021, 39:100317.

24. Polat H. and Turkoglu M. and Polat O. and Sengur A., A novel approach for accurate detection of the DDoS attacks in SDN-based SCADA systems based on deep recurrent neural networks, Expert Systems With Applications, 2022, 197:116748.

25. Shaukat K. and Luo S. and Varadharajan V., A novel method for improving the robustness of deep learning-based malware detectors against adversarial attacks, Engineering Applications of Artificial Intelligence, 2022, 116:105461.

CONFLICTO DE INTERESES

No existe conflicto de intereses entre los autores, ni con la institución a la que están afiliados, ni con ninguna otra institución.

CONTRIBUCIONES DE LOS AUTORES

Adrián Rodríguez-Ramos: Conceptualización, Curación de datos, Análisis formal, Investigación, Metodología, Software, Validación-Verificación, redacción-Borrador Original

Orestes Llanes-Santiago: Conceptualización, Análisis formal, Adquisición de fondos, Investigación, Metodología, Administración de Proyecto, Recursos, Supervisión, Validación-Verificación, Redacción-revisión y edición.

AUTORES

Adrián Rodríguez Ramos, Ingeniero en Automática. Máster en Modelación Matemática Aplicada a la Ingeniería. Doctorado en Ciencias Técnicas, Universidad Tecnológica de La Habana José Antonio Echeverría (CUJAE), La Habana, Cuba. Email: adrianrr@automatica.cujae.edu.cu, No ORCID 0000-0002-0240-7491.Sus intereses de investigación se orientan hacia la inteligencia artificial y el diagnóstico de fallos en proceso industriales.

Orestes Llanes-Santiago, Ingeniero Electricista. Doctor en Ciencias Aplicadas. Profesor e Investigador Titular de la Universidad Tecnológica José Antonio Echeverría (CUJAE). Académico Titular de la Academia de Ciencias de Cuba. Email: orestes@tesla.cujae.edu.cu No. ORCID 0000-0002-6864-9629. Sus principales intereses de investigación están en el monitoreo de condición y diagnóstico de fallos en sistemas industriales, implementación del paradigma Industria 4.0, inteligencia computacional aplicada al control y el control de sistemas no lineales.