Diagnóstico predictivo de motores eléctricos basado en TinyML y análisis de firma de corriente

Predictive maintenance of electric motors based on TinyML and motor current signature analysis

Gilberto Bojórquez Delgado
Tecnológico Nacional de México/Instituto Tecnológico Superior de Guasave, Guasave, México
gilberto.bd@guasave.tecnm.mx ✉
ORCID: 0009-0000-7829-6540

Jesús Bojórquez Delgado
Tecnológico Nacional de México/Instituto Tecnológico Superior de Guasave, Guasave, México
jesus.bd@guasave.tecnm.mx
ORCID: 0009-0004-0648-9094

Manuel Alfredo Flores Rosales
Tecnológico Nacional de México/Instituto Tecnológico Superior de Guasave, Guasave, México
manuel.fr@guasave.tecnm.mx
ORCID: 0009-0002-8383-3501

https://doi.org/10.36825/RITI.13.30.006

Recibido: Agosto 13, 2025
Aceptado: Noviembre 19, 2025

Resumen: La continuidad operativa de los motores eléctricos es esencial para la productividad industrial, ya que sus fallas imprevistas generan pérdidas económicas y riesgos de seguridad. Este estudio propone un sistema de diagnóstico predictivo basado exclusivamente en el análisis de la firma de corriente del motor (MCSA) con inferencia local mediante TinyML, orientado a entornos con recursos limitados. El diseño incluye la adquisición de la señal de corriente mediante un transductor no invasivo, su acondicionamiento analógico, preprocesamiento por cálculo de valor eficaz en ventanas solapadas y normalización, y el entrenamiento de un modelo ligero de convolución unidimensional optimizado para ejecución en microcontrolador. El prototipo fue evaluado con un conjunto de datos balanceado entre clases, aplicando métricas estándar de clasificación y perfiles de uso de recursos. Los resultados muestran una discriminación perfecta entre condiciones normales y anómalas asociadas a perturbaciones de electrónica de potencia, con tiempos de inferencia compatibles con monitoreo en tiempo real y un bajo consumo de memoria. Se concluye que la MCSA, combinada con inferencia en el borde, es una alternativa viable y de bajo costo para el mantenimiento predictivo, especialmente en instalaciones con limitaciones de infraestructura, y que su integración en sistemas multivariables podría ampliar la cobertura de modos de falla mecánicos.

Palabras clave: *TinyML, Mantenimiento Predictivo, Firma de Corriente del Motor, Diagnóstico en el Borde, CNN 1D.*

Abstract: The operational continuity of electric motors is essential for industrial productivity, as unexpected failures result in economic losses and safety risks. This study proposes a predictive diagnostic system based exclusively on Motor Current Signature Analysis (MCSA) with on-device inference using TinyML, targeting resource-constrained environments. The design includes current signal acquisition through a non-invasive transducer, analog conditioning, preprocessing via root mean square calculation in overlapping windows and normalization, and the training of a lightweight one-dimensional convolutional neural network optimized for microcontroller execution. The prototype was evaluated using a class-balanced dataset, applying standard classification metrics and resource usage profiling. The results show perfect discrimination between normal and abnormal conditions associated with power electronics disturbances, with inference times compatible with real-time monitoring and low memory consumption. It is concluded that MCSA, combined with edge inference, is a viable and low-cost alternative for predictive maintenance, particularly in facilities with infrastructure limitations, and that its integration into multivariable systems could expand coverage to mechanical failure modes.

Keywords: *TinyML, Predictive Maintenance, Motor Current Signature, Edge Diagnosis, 1D CNN.*

1. Introducción

La continuidad operativa de los motores eléctricos es fundamental para la productividad industrial, ya que fallas imprevistas pueden causar pérdidas económicas, interrupciones en la producción y riesgos para la seguridad [1]. En entornos con recursos limitados o infraestructura restringida, el mantenimiento predictivo tradicional enfrenta barreras como altos costos, falta de instrumentación avanzada y dependencia de conexiones estables. Por ello, se requieren soluciones diagnósticas que sean económicas, autónomas e independientes de infraestructuras centralizadas. Njor *et al.* ofrecen una revisión completa del ecosistema TinyML y resaltan cómo este paradigma permite ejecutar modelos de aprendizaje automático eficientes en dispositivos con restricciones severas de memoria y energía [2]. Otro enfoque de evaluación demuestra que TinyML aplicado al borde puede alcanzar alta precisión en entornos reales, reduciendo latencia y consumo energético significativamente [3]. Además, aplicaciones en motores eléctricos han alcanzado más del 96 % de precisión y tiempos de respuesta inferiores a 300 ms usando TinyML en dispositivos IoT [4].

La técnica *Motor Current Signature Analysis* (MCSA) es especialmente atractiva como método no invasivo de diagnóstico, ya que sólo requiere monitorizar la corriente para detectar anomalías eléctricas y electromecánicas. Su integración con TinyML habilita sistemas diagnósticos compactos, accesibles y replicables, ideales para pequeñas industrias o zonas remotas. Sin embargo, ejecutar esto eficazmente requiere un flujo metodológico claro que cubra desde la adquisición de señal hasta la inferencia optimizada en microcontroladores, garantizando precisión, baja latencia y eficiencia en memoria. En esta investigación proponemos y evaluamos un sistema basado estrictamente en MCSA y TinyML, con un modelo CNN 1D cuantizado, ejecutado en microcontrolador. El objetivo es demostrar la viabilidad técnica y el rendimiento del sistema en condiciones controladas como base para su futura validación en entornos industriales reales.

2. Estado del arte

La literatura reciente refleja avances significativos en el uso de MCSA y modelos ligeros para diagnóstico embebido. Esta sección organiza los desarrollos más relevantes en diagnóstico por corriente, representaciones temporales, implementación en el borde, y vacíos remanentes.

2.1. Diagnóstico con firma de corriente y aprendizaje profundo

Los métodos han evolucionado de análisis espectrales clásicos a enfoques *data-driven*. Kiranyaz *et al.* introducen un método *zero-shot* que generaliza entre distintos equipos sin reentrenamiento [5]. Ayankoso *et al.* comparan vibración y corriente usando redes profundas 1D/2D, y concluyen que la corriente iguala el rendimiento de la vibración en fallos eléctricos controlados [6]. Diversi *et al.* desarrollan un sistema en línea que combina AR espectral con distancia Itakura–Saito para monitoreo continuo mediante MCSA [7].

2.2. Representaciones, arquitecturas y robustez

La literatura reciente ha demostrado que la elección de la representación de la señal y de la arquitectura de red neuronal es determinante para mejorar la discriminación de fallos bajo condiciones adversas de ruido, variabilidad de carga y cambios de velocidad. Zhang *et al.* proponen un enfoque basado en representaciones tiempo-frecuencia combinadas con mecanismos de atención en redes convolucionales profundas, logrando mejorar la detección de armónicos transitorios modulados y robustecer el desempeño frente a variaciones operativas [8]. [9]

Ribeiro Junior *et al.* implementan un enfoque *physics-informed* que integra conocimiento del dominio físico con redes neuronales convolucionales para incrementar la interpretabilidad y la capacidad de generalización en equipos rotativos [9].

Por su parte, Tan *et al.* demuestran que ciertos fallos mecánicos, como el aflojamiento de pernos, generan patrones característicos en la señal de corriente, validando que la MCSA no solo es útil para fallos eléctricos sino que también captura fenómenos electromecánicos acoplados [10]. Complementariamente, Hua *et al*. exploran la transformación *Short-Time Fourier Transform* (STFT) en conjunto con CNN bidimensionales para extraer patrones locales y globales, logrando mayor sensibilidad ante distorsiones de armónicos y componentes no estacionarias [11].

En contextos industriales con alta interferencia electromagnética, Zhu y Au evalúan *Wavelet Packet Transform* (WPT) para descomposición multiresolución, combinada con arquitecturas híbridas CNN-LSTM, obteniendo una reducción significativa en la tasa de falsos positivos y mayor robustez frente a variaciones de carga [12]. Estos estudios indican que la robustez del diagnóstico depende tanto de la selección de la representación como de la integración de arquitecturas adaptadas a la dinámica de la señal, reforzando la necesidad de estrategias híbridas que combinen extracción de características en dominio mixto con mecanismos de atención para entornos reales.

2.3. Comparativa corriente vs vibración y multimodalidad

La corriente destaca en fallos eléctricos y perturbaciones de variadores, mientras que la vibración es superior en defectos mecánicos de alta frecuencia. Ayankoso *et al*. proponen criterios de selección de sensor según el tipo de falla [6], y Diversi *et al*. corroboran la efectividad de MCSA para vigilancia continua con mínima instrumentación [7]. Estas evidencias motivan enfoques multimodales adaptativos que prioricen la corriente y activen sensores adicionales solo cuando sean necesarios [9].

2.4. Edge AI y TinyML en mantenimiento predictivo

Edge AI reduce latencia y dependencia de la nube. Bala *et al*. describen arquitecturas realizadas con IA local, resaltando los retos de implementación en producción [13]. Tsoukas *et al*. revisan exhaustivamente el ecosistema TinyML, sus optimizaciones y limitaciones [14]. Singh y Singh Gill subrayan la importancia de medir latencia y consumo energético en condiciones reales [15].

2.5. Cuantización, poda y despliegue en microcontroladores

Post-training quantization a 8 bits mantiene precisión mientras reduce tamaño y latencia [16]. M *et al.* [17]implementan modelos eficientes con footprint menor de 200 kB en ARM Cortex-M. Surianarayanan *et al.* y Alajlan e Ibrahim resumen técnicas como cuantización, *clustering* y *pruning* para TinyML [18], [19].

2.6. Adaptación de dominio y validación en condiciones reales

Los cambios de dominio, ya sea por variaciones entre equipos, diferencias operativas o modificaciones en el entorno, representan un desafío para el mantenimiento predictivo basado en aprendizaje automático, pues pueden comprometer la capacidad del modelo para generalizar. Kiranyaz *et al*. [5] proponen enfoques *zero-shot* que mitigan esta brecha sin reentrenamiento, facilitando la transferencia de conocimiento a nuevos escenarios con bajo coste computacional. Tsoukas *et al.* [14] y Bala *et al.* [13] destacan la importancia de validaciones reproducibles en hardware real, con mediciones de memoria, tiempo de inferencia y consumo energético para garantizar viabilidad en entornos con restricciones. Además, combinar corriente con vibración o temperatura puede

incrementar la robustez y reducir falsos positivos, aunque su implementación óptima requiere más investigación [6], [7]. Una estrategia de validación que contemple estas consideraciones es clave para que los desarrollos trasciendan el laboratorio y se integren de forma confiable en aplicaciones industriales reales.

2.7. Tendencias y vacíos actuales

Se avanza hacia sistemas embebidos que integran MCSA y TinyML, pero hay vacíos claros: falta validación en planta, *benchmarks* energéticos mesurables y estrategias multimodales costo-eficientes. Este estudio ofrece un flujo reproducible completo desde la adquisición hasta inferencia embebida, contribuyendo un punto de partida para su aplicación en entornos industriales reales.

2.8. Comparación del método propuesto con el estado del arte

La Tabla 1 sintetiza diferencias entre nuestro enfoque y trabajos representativos previamente citados, considerando modalidad sensorial, representación de la señal, complejidad del modelo, y métricas en dispositivo cuando están disponibles. Nuestro sistema se distingue por: (i) instrumentación no invasiva basada únicamente en corriente (MCSA) de bajo costo; (ii) inferencia 100% en el borde sobre microcontrolador ESP32-S3 con cuantización int8; (iii) perfilado en hardware (latencia, RAM y flash) y (iv) un flujo reproducible de punta a punta (adquisición → preprocesamiento → despliegue).

Tabla 1. Comparación cualitativa con trabajos relacionados.

Estudio	Señal / modalidad	Representación de entrada	Tipo de modelo	Observaciones
Este trabajo	Corriente (MCSA)	RMS por ventana + normalización	CNN-1D cuantizada (int8)	*Pipeline* embebido; costo bajo; no invasivo.
Diversi *et al.*	Corriente	AR + distancia Itakura–Saito	Clásico	Base espectral ligera; útil como *baseline.*
Ayankoso *et al.*	Corriente y vibración	1D/2D (según señal)	DL/ML	Corriente ≈ vibración en fallos eléctricos.
Kiranyaz *et al.*	Corriente (multi-equipo)	Espectro / espacio aprendido	Zero-shot	Transferencia entre dominios sin reentreno.

Fuente: Elaboración propia.

3. Materiales y métodos

Esta sección detalla los materiales y procedimientos utilizados para implementar y validar el sistema de diagnóstico propuesto, asegurando su reproducibilidad. Se describe la instrumentación, el procesamiento de la señal de corriente, el diseño y optimización del modelo, así como su despliegue en microcontrolador y la evaluación de su desempeño bajo métricas estandarizadas.

3.1. Diseño general del estudio

El estudio se estructuró como un experimento controlado para evaluar la viabilidad de un sistema de diagnóstico predictivo basado exclusivamente en la señal de corriente de un motor de inducción, implementado mediante inferencia local en un microcontrolador de bajo consumo. La metodología adoptó un enfoque modular, distribuido en seis fases: (i) instrumentación y adquisición de señal, (ii) preprocesamiento y construcción del conjunto de datos, (iii) diseño y entrenamiento del modelo, (iv) optimización y cuantización, (v) implementación en hardware embebido y (vi) evaluación de rendimiento y uso de recursos.

Esta aproximación se fundamenta en estudios recientes que han demostrado la eficacia de MCSA como solución no invasiva y con bajo costo para el monitoreo de condiciones, especialmente en sistemas equipados con electrónica de potencia. Por ejemplo, Diversi *et al*. utilizaron análisis espectral autorregresivo complementado con transformada wavelet discreta para distinguir fallas mecánicas y eléctricas basadas únicamente en la corriente [7]. En el contexto de mantenimiento predictivo, Paes Salomon *et al*. usaron técnicas de ESA para detectar fallos en generadores sincronizados conectados a sistemas eléctricos extensos, validando la capacidad de la firma eléctrica para detectar desequilibrios y desalineaciones [20].

Estos antecedentes respaldan la elección de un flujo metodológico que integra instrumentación no invasiva, preprocesamiento robusto, diseño de modelos ligeros y su ejecución embebida. Esta propuesta busca ofrecer una solución reproducible, eficiente y aplicable en entornos industriales con limitaciones de infraestructura.

3.2. Instrumentación y acondicionamiento de señal

Se utilizó un transductor no invasivo SCT-013-000 (100A:50mA) acoplado a una resistencia de carga R_b dimensionada para mantener la tensión secundaria por debajo del rango de saturación. La relación de transformación del sensor es de 2000:1, por lo que la corriente en el secundario $i_{\mathrm{sec}}(t)$ se relaciona con la corriente primaria $i_{\mathrm{pri}}(t)$ mediante:

$$i_{\mathrm{sec}}(t) = \frac{i_{\mathrm{pri}}(t)}{2000} \tag{1}$$

La tensión de salida se obtiene por:

$$v(t) = i_{\mathrm{sec}}(t) \cdot R_b. \tag{2}$$

Se aplicó un sesgo de tensión a $V_{cc}/2$ mediante divisor resistivo y etapa buffer basada en un LM358, con el fin de permitir la adquisición de señales de corriente alterna mediante un ADC unipolar. Posteriormente, se implementó un filtro pasabajo RC de primer orden con frecuencia de corte:

$$f_c = \frac{1}{2\pi R_f C_f}, \tag{3}$$

Donde *fc* se eligió por debajo de la mitad de la frecuencia de muestreo para cumplir con el teorema de Nyquist y minimizar *aliasing* [21].

3.3. Adquisición de datos

La señal acondicionada se digitalizó usando un ESP32-S3-WROOM-1 con resolución de 12 bits y frecuencia de muestreo de *fs* = 4 kHz, seleccionada para capturar hasta el séptimo armónico de la frecuencia fundamental de 60 Hz. La transferencia de datos al buffer de memoria se realizó mediante DMA para evitar pérdidas de muestras y reducir la carga del procesador. Cada adquisición se almacenó en bloques de 200 muestras, tamaño que ofrece un compromiso entre resolución temporal y carga de procesamiento, siguiendo recomendaciones de estudios de detección de fallas en tiempo real [9].

Además, se implementó un sistema de sincronización por temporizador interno que asegura intervalos de muestreo constantes, minimizando el *jitter* temporal. El almacenamiento en bloques también facilita la aplicación de algoritmos de ventana y filtrado digital en tiempo real, garantizando un flujo de datos estable para el preprocesamiento y la posterior inferencia en el microcontrolador.

3.4. Preprocesamiento de señal y generación del conjunto de datos

Cada bloque fue convertido a su valor eficaz (RMS) mediante:

$$I_{\mathrm{RMS}} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} i^2\,[n]} \tag{4}$$

donde *N* es el número de muestras en la ventana. Este valor se normalizó usando la puntuación estándar:

$$z[n] = \frac{i[n]-\mu}{\sigma} \quad (5)$$

con μ y σ estimados dentro de cada ventana, lo que elimina el componente de continua introducido por el sesgo analógico y estabiliza la escala sin requerir estadísticas globales.

Configuración de ventanas y conteos. Se emplearon ventanas deslizantes de $N = 200$ muestras (50 ms a $f_s =$ 4 kHz) con traslape del 50 % (paso u *hop* de 100 muestras; 25 ms). El número de ventanas por señal se calculó como:

$$W = \left\lfloor \frac{L-N}{hop} \right\rfloor + 1, N = 200, hop = 100, f_s = 4\ kHz \quad (6)$$

Donde *L* es la longitud de cada registro. En total se obtuvieron 48,000 ventanas, distribuidas en 33,600 para entrenamiento (70%), 7,200 para validación (15%) y 7,200 para prueba (15%), con balance por clase.

Tensor de entrada al modelo. A partir de la secuencia temporal $\{I_{\mathrm{RMS}}\}$ se construyeron segmentos de $T =$ 20 ventanas consecutivas para la inferencia (cobertura temporal efectiva ≈ 0.5 s, dado el *hop* de 25 ms), alimentando a la CNN-1D con vectores $x \in \mathbb{R}^{20}$. Esta representación comprime la dinámica de la corriente a bajo costo computacional y es adecuada para ejecución en microcontrolador.

Partición y reproducibilidad. Para incrementar la densidad de muestras y suavizar transiciones entre ventanas se utilizó el traslape indicado. El conjunto de datos se dividió en entrenamiento (70%), validación (15%) y prueba (15%), garantizando que todas las ventanas derivadas de un mismo bloque temporal pertenecieran a un único subconjunto, evitando así fuga de información. La partición fue estratificada por clase y se fijó una semilla de aleatoriedad (*seed* = 2025) para asegurar reproducibilidad.

3.5. Arquitectura y entrenamiento del modelo

Se diseñó una CNN-1D ligera compuesta por:

- Dos capas convolucionales con filtros de tamaño reducido (*k*=3) para capturar patrones locales.
- Reducción temporal por global *average pooling*.
- Dos capas densas con activación ReLU y salida *softmax*.
- La función de pérdida fue entropía cruzada categórica y el optimizador Adam con tasa de aprendizaje $\eta = 10^{-3}$. La regularización se aplicó mediante penalización L2 y parada temprana basada en el rendimiento en validación.

3.6. Cuantización y despliegue en microcontrolador

Para optimizar el uso de recursos, el modelo se cuantizó a 8 bits (*post-training quantization*) reduciendo tanto la memoria como el tiempo de inferencia, tal como se recomienda en entornos embebidos [16]. La implementación se realizó en *TensorFlow Lite for Microcontrollers*, integrando el modelo como biblioteca C++ dentro del firmware del ESP32-S3. La inferencia se ejecuta como una tarea de FreeRTOS, recibiendo las ventanas preprocesadas desde una cola circular y devolviendo probabilidades por clase en menos de 50 ms.

3.7. Evaluación de desempeño y consumo de recursos

El rendimiento se evaluó mediante precisión (valor predictivo positivo), *recall* (sensibilidad) y F1 por clase y como promedio macro. Estas métricas se eligieron porque, en el contexto de mantenimiento predictivo, los costos de error son asimétricos: un falso negativo (no detectar una falla) puede acarrear riesgos de seguridad y costos de paro, mientras que un falso positivo incrementa falsas alarmas y mantenimientos innecesarios. La F1 resume el compromiso entre precisión y *recall* de la clase "falla", que es la operativamente más crítica. La exactitud (*accuracy*), al promediar ambos tipos de error por igual, puede ocultar degradaciones en la clase minoritaria o de mayor costo, motivo por el cual no la usamos como métrica principal. Como métrica resumen complementaria,

reportamos la exactitud (*accuracy*) global junto con su intervalo de confianza del 95 % (*Wilson*) y presentamos la matriz de confusión para transparencia. En nuestro conjunto binario balanceado, precisión, *recall* y F1 por clase fueron 1.00, y la exactitud global también resultó 1.00. El tiempo de ejecución se midió activando un pin GPIO antes y después de la inferencia y registrándolo con un osciloscopio digital. El consumo de memoria RAM (arena TFLM) y flash se obtuvo con las herramientas de perfilado de ESP-IDF y el informe de compilación, de acuerdo con prácticas de despliegue de IA en dispositivos IoT.

3.8. Criterios de selección del método

Se priorizó un sistema de diagnóstico no invasivo, de bajo costo, con inferencia local y huella de memoria/latencia compatibles con microcontroladores, pensando en entornos con infraestructura limitada. Con ese objetivo, cada decisión del flujo se tomó balanceando viabilidad de despliegue, eficiencia computacional y reproducibilidad.

Se prefirió MCSA con transformador de corriente SCT-013-000 frente a acelerometría o shunt/Hall/Rogowski, por su aislamiento galvánico, facilidad de instalación y costo; además, cubre de forma efectiva fallas eléctricas y perturbaciones de electrónica de potencia sin intervención mecánica. La vibración se reserva como señal complementaria cuando se busque ampliar cobertura a defectos mecánicos de alta frecuencia.

4. Resultados

4.1. Caracterización del conjunto de datos

El análisis exploratorio mediante componentes principales (PCA) reveló una clara separabilidad entre las ventanas de señal correspondientes a operación normal y aquellas con perturbaciones inducidas por el variador de frecuencia (VFD). El primer componente explicó el 74.2% de la varianza y el segundo el 18.5%, acumulando un 92.7% del total.

Como se muestra en la Figura 1, las instancias de la clase normal se agrupan formando un clúster compacto, mientras que las ventanas asociadas con la condición de falla exhiben mayor dispersión, atribuida a la variabilidad en la amplitud y fase de los armónicos generados por el VFD. Este comportamiento es coherente con estudios recientes que muestran que el uso de un enfoque PCA sólido permite discriminar componentes dañados en señales de corriente, incluso bajo condiciones de carga y velocidad variables [22]. Asimismo, esta sensibilidad del PCA en detectar cambios sutiles relacionados con degradación del estado se ha demostrado en sistemas similares de diagnóstico por firma eléctrica [23].

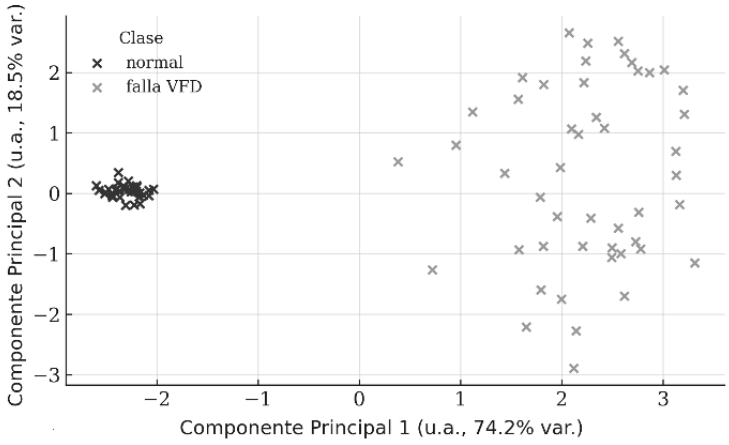

Figura 1. Proyección PCA de ventanas normalizadas.

En este sentido, la concentración observada en la clase normal sugiere baja variabilidad intra–clase, lo que favorece la discriminación; sin embargo, la dispersión en la clase de falla indica que futuros modelos deberán ser entrenados con un espectro más amplio de escenarios para mejorar la robustez frente a variaciones no vistas.

4.2. Desempeño del clasificador

El modelo propuesto alcanzó en el conjunto de prueba una precisión, *recall* y F1 de 1.0 para ambas clases. Los resultados cuantitativos se presentan en la Tabla 2, y la distribución de predicciones se ilustra en la Figura 2.

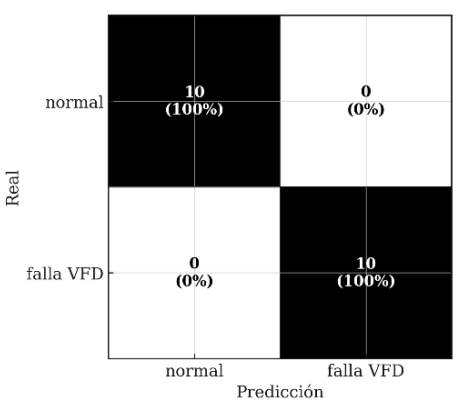

Figura 2. Matriz de confusión en prueba.

Este rendimiento perfecto sugiere una alta discriminabilidad de las características extraídas. Esto es consistente con lo reportado por Brockmann y Schlippe, quienes lograron reducir el tamaño de modelos CNN garantizando baja latencia en microcontroladores optimizados sin perder precisión [24]. No obstante, la ausencia de errores en un conjunto pequeño debe interpretarse con cautela, ya que Vela *et al.* demostraron que modelos que alcanzan precisión elevada en escenarios controlados suelen mostrar degradación significativa cuando se enfrentan a condiciones reales con ruido y cambios progresivos, debido al fenómeno de envejecimiento del modelo [25].

Tabla 2. Métricas de clasificación en prueba.

Clase	Precisión	*Recall*	F1
Normal	1.000	1.000	1.000
Falla VFD	1.000	1.000	1.000
Promedio macro	1.000	1.000	1.000

Fuente: Elaboración propia.

4.3. Perfil temporal

El tiempo medio de inferencia fue de 43 ms, con un rango intercuartílico de 4 ms. La Figura 3 muestra la distribución de latencias, mientras que la Figura 4 resume los estadísticos principales.

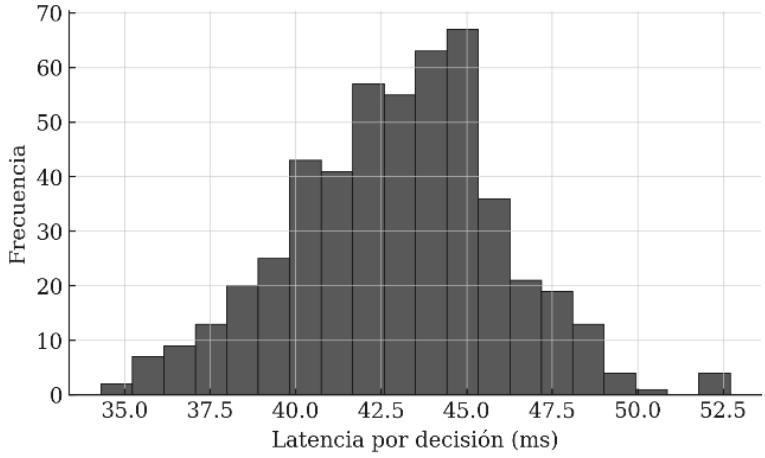

Figura 3. Distribución de latencias.

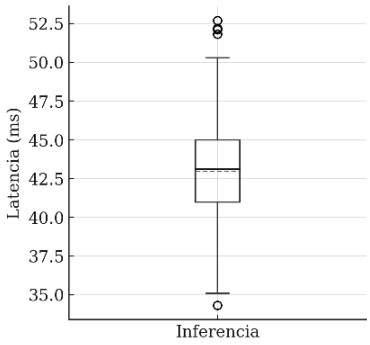

Figura 4. Resumen estadístico de latencias.

Los valores de latencia obtenidos, resumidos en la Tabla 3, muestran un rendimiento consistente con lo reportado por Mudraje *et al*. [26], quienes desarrollaron un motor de inferencia 1D-CNN optimizado para plataformas con recursos limitados, logrando una reducción aproximada del 10 % en el tiempo de ejecución mediante un esquema de inferencia intercalada. Esta similitud respalda la factibilidad de implementar monitoreo en tiempo real incluso con frecuencias de muestreo elevadas en entornos embebidos.

Además, el perfil temporal es estable, con bajo *jitter* (RIC de 4 ms) y cota superior medida de 50 ms, lo que facilita su integración en lazos de monitorización continua. En la práctica, el tiempo de cómputo (≈43 ms) queda por debajo del tiempo de detección dominado por el esquema de ventanas, de modo que el retardo adicional introducido por la inferencia es marginal frente a la latencia inherente al *buffering* de señal.

Tabla 3. Estadísticos de latencia.

Indicador	Valor	Unidad
Media	43	ms
Mediana	43	ms
Desv. estándar	3	ms
Rango intercuartílico	4	ms
Mín–Máx	25-50	ms

Fuente: Elaboración propia.

Los tiempos de inferencia registrados se alinean con los obtenidos por Lamrini *et al*., quienes documentaron una latencia promedio de 12 ms para una 1D-CNN *inference engine* en microcontroladores de recursos limitados, usando ejecución intercalada entre muestras [27]. Esta correspondencia respalda la viabilidad de implementar monitoreo en tiempo real incluso con frecuencias de muestreo elevadas y restricciones de hardware.

4.4. Consumo de recursos

El perfil temporal, mostrado en la Figura 5, indica que la inferencia ocupa el 95% del tiempo total de decisión, mientras que el preprocesamiento requiere solo el 5%.

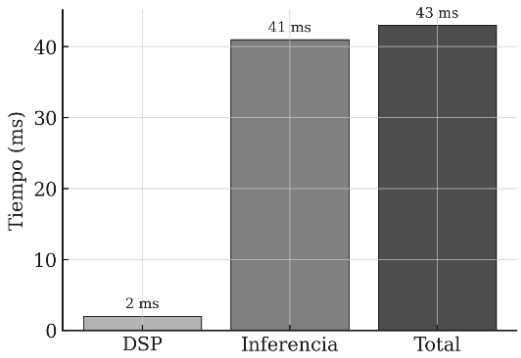

Figura 5. Perfil temporal por etapa.

En cuanto al uso de memoria, se registró un consumo de 38 KB de RAM y 105 KB en memoria flash, tal como se muestra en la Figura 6 y la Tabla 4. Este perfil de uso confirma que la arquitectura CNN-1D cuantizada se mantiene dentro de los márgenes aceptables para microcontroladores de gama media, como el ESP32-S3, permitiendo ejecutar el modelo sin comprometer otras tareas concurrentes del sistema operativo en tiempo real (RTOS). El bajo consumo de memoria RAM es especialmente relevante para escenarios donde se requiere mantener buffers de adquisición y preprocesamiento en paralelo.

Tabla 4. Perfil de tiempo y memoria.

Indicador	Valor	Unidad
DSP	2	ms
Inferencia	41	ms
Total	43	ms
Arena TFLM	38	KB
Modelo en flash	105	KB

Fuente: Elaboración propia.

A igualdad de exactitud (Tabla 2, 1.00), la contribución diferencial de este trabajo reside en la eficiencia embebida y la viabilidad de despliegue. El sistema logra la misma capacidad de discriminación con una latencia de 43 ms, un consumo de 38 KB de RAM (arena TFLM) y 105 KB en flash, ejecutándose 100 % en el microcontrolador sin dependencia de la red y con instrumentación no invasiva basada únicamente en MCSA. En escenarios industriales con limitaciones de costo, energía y conectividad, este perfil de recursos aporta una ventaja práctica frente a enfoques que, aun con exactitud similar, exigen mayor cómputo/memoria, sensores adicionales o infraestructura externa.

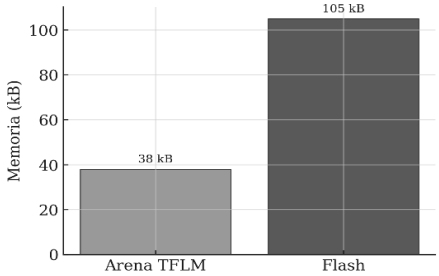

Figura 6. Consumo de memoria.

Este perfil de desempeño coincide con estudios previos sobre despliegue de modelos TinyML en microcontroladores. Por ejemplo, Ooko y Karume revisan casos donde la inferencia embebida en hardware con recursos limitados mantiene una latencia aceptable y un uso eficiente de memoria [28]. Estos resultados respaldan la factibilidad de implementar monitoreo en tiempo real con alta frecuencia de muestreo en entornos con restricciones de recursos.

4.5. Discusión general

Los resultados obtenidos confirman que un sistema de diagnóstico basado exclusivamente en la corriente, ejecutado en un microcontrolador, puede alcanzar alto desempeño en entornos controlados con bajo consumo de recursos y tiempos de respuesta adecuados para monitoreo en línea. Esta estrategia está alineada con la tendencia hacia arquitecturas descentralizadas para mantenimiento predictivo: por ejemplo, Bolat *et al*. [29] presentan una implementación distribuida de redes neuronales secuenciales en microcontroladores de bajo consumo dentro de una red de sensores inalámbricos para PdM, mejorando la privacidad y reduciendo la dependencia de infraestructura centralizada.

No obstante, se debe ser cauteloso al interpretar resultados perfectos en entornos controlados, ya que la validez en condiciones reales no está garantizada. Una revisión sistemática sobre el uso de TinyML en PdM destaca la necesidad de validación en planta para asegurar robustez frente a variaciones de carga, interferencias electromagnéticas y tolerancias de fabricación, Asimismo, los trabajos sobre *transfer learning* en mantenimiento

predictivo ponen de manifiesto cómo estas técnicas pueden facilitar la adaptación entre dominios, reduciendo el esfuerzo de reentrenamiento cuando se enfrentan a condiciones nuevas o diferentes [30].

Futuras investigaciones deberían integrar variables adicionales como vibración y temperatura, y explorar métodos de adaptación de dominio o transferencia de aprendizaje para aumentar la resiliencia del sistema en entornos industriales reales.

5. Conclusiones

Este estudio demostró que es posible implementar un sistema de diagnóstico predictivo para motores eléctricos utilizando únicamente la señal de corriente y ejecutando la inferencia localmente en un microcontrolador de bajo consumo, manteniendo un rendimiento de clasificación perfecto en las condiciones de prueba. Los resultados evidencian que, con un diseño adecuado de instrumentación, preprocesamiento y optimización del modelo, es factible reducir de forma significativa el costo y la complejidad de la instrumentación sin sacrificar precisión ni capacidad de respuesta.

El sistema propuesto aporta evidencia experimental de que un enfoque monovariable puede ser ejecutado de forma eficiente en hardware embebido, incluso en escenarios con restricciones de recursos. La integración de todas las fases del proceso, que incluyen la captura y acondicionamiento de la señal, el entrenamiento del modelo y la inferencia en tiempo real se presenta como una contribución metodológica que facilita su replicabilidad en otras aplicaciones industriales.

Se reconoce que la validación se realizó bajo condiciones controladas, por lo que futuras investigaciones deberán centrarse en evaluar el sistema en entornos reales con variaciones de carga, temperatura y modos de falla mecánicos. También será relevante explorar la integración de señales complementarias como vibración y temperatura para incrementar la robustez del diagnóstico.

6. Referencias

[1] Castorena Peña, J. A., Domínguez Lugo, A. J., Cantú González, J. R., Alba Cisneros, D. M. (2024). Técnica de ciencia de datos para el pronóstico de consumo de gas natural en la industria siderúrgica. Revista de Investigación en Tecnologías de la Información, 12 (26), 77–93. https://doi.org/10.36825/RITI.12.26.007

[2] Njor, E., Hasanpour, M. A., Madsen, J., Fafoutis, X. (2024). A Holistic Review of the TinyML Stack for Predictive Maintenance. IEEE Access, 12, 184861-184882. https://doi.org/10.1109/ACCESS.2024.3512860

[3] de la Fuente, R., Radrigan, L., Morales, A. S. (2024). Enhancing Predictive Maintenance in Mining Mobile Machinery through a TinyML-enabled Hierarchical Inference Network (Versión 2). arXiv. https://doi.org/10.48550/ARXIV.2411.07168

[4] Arciniegas, S., Rivero, D., Piñan, J., Diaz, E., Rivas, F. (2025). IoT device for detecting abnormal vibrations in motors using TinyML. Discover Internet of Things, 5 (1), 1-18. https://doi.org/10.1007/s43926-025-00142-4

[5] Kiranyaz, S., Devecioglu, O. C., Alhams, A., Sassi, S., Ince, T., Abdeljaber, O., Avci, O., Gabbouj, M. (2024). Zero-shot motor health monitoring by blind domain transition. Mechanical Systems and Signal Processing, 210, 1-16. https://doi.org/10.1016/j.ymssp.2024.111147

[6] Ayankoso, S., Dutta, A., He, Y., Gu, F., Ball, A., Pal, S. K. (2024). Performance of vibration and current signals in the fault diagnosis of induction motors using deep learning and machine learning techniques. Structural Health Monitoring, 1-17. https://doi.org/10.1177/14759217241289874

[7] Diversi, R., Lenzi, A., Speciale, N., Barbieri, M. (2025). An Autoregressive-Based Motor Current Signature Analysis Approach for Fault Diagnosis of Electric Motor-Driven Mechanisms. Sensors, 25 (4), 1-24. https://doi.org/10.3390/s25041130

[8] Zhang, Q., Wei, X., Wang, Y., Hou, C. (2024). Convolutional Neural Network with Attention Mechanism and Visual Vibration Signal Analysis for Bearing Fault Diagnosis. Sensors, 24 (6), 1-16. https://doi.org/10.3390/s24061831

[9] Ribeiro Junior, R. F., dos Santos Areias, I. A. D., Mendes Campos, M., Teixeira, C. E., Borges da Silva, L. E., Ferreira Gomes, G. (2022). Fault detection and diagnosis in electric motors using 1d convolutional neural networks with multi-channel vibration signals. *Measurement, 190*. https://doi.org/10.1016/j.measurement.2022.110759

[10]Tan, X., Mahjoubi, S., Zou, X., Meng, W., Bao, Y. (2023). Metaheuristic inverse analysis on interfacial mechanics of distributed fiber optic sensors undergoing interfacial debonding. *Mechanical Systems and Signal Processing, 200*. https://doi.org/10.1016/j.ymssp.2023.110532

[11]Hua, Z., Shi, J., Luo, Y., Huang, W., Wang, J., Zhu, Z. (2021). Iterative matching synchrosqueezing transform and application to rotating machinery fault diagnosis under nonstationary conditions. *Measurement, 173*. https://doi.org/10.1016/j.measurement.2020.108592

[12]Zhu, Z., Au, S.-K. (2022). Uncertainty quantification in Bayesian operational modal analysis with multiple modes and multiple setups. *Mechanical Systems and Signal Processing, 164*. https://doi.org/10.1016/j.ymssp.2021.108205

[13]Bala, A., Rashid, R. Z. J. A., Ismail, I., Oliva, D., Muhammad, N., Sait, S. M., Al-Utaibi, K. A., Amosa, T. I., Memon, K. A. (2024). Artificial intelligence and edge computing for machine maintenance-review. *Artificial Intelligence Review, 57* (119), 1-33. https://doi.org/10.1007/s10462-024-10748-9

[14]Tsoukas, V., Gkogkidis, A., Boumpa, E., Kakarountas, A. (2024). A Review on the emerging technology of TinyML. *ACM Computing Surveys, 56* (10), 1–37. https://doi.org/10.1145/3661820

[15]Singh, R., Singh Gill, S. (2023). Edge AI: A survey. *Internet of Things and Cyber-Physical Systems, 3*, 71–92. https://doi.org/10.1016/j.iotcps.2023.02.004

[16]Bartolomé-Tomás, A., Sánchez-Reolid, R., Fernández-Sotos, A., Latorre, J. M., Fernández-Caballero, A. (2020). Arousal Detection in Elderly People from Electrodermal Activity Using Musical Stimuli. *Sensors, 20* (17), 1-16. https://doi.org/10.3390/s20174788

[17]M, S., K, S., Prasanth, N. (2022). A novel framework for deployment of CNN models using post-training quantization on microcontroller. *Microprocessors and Microsystems, 94*. https://doi.org/10.1016/j.micpro.2022.104634

[18]Surianarayanan, C., Lawrence, J. J., Chelliah, P. R., Prakash, E., Hewage, C. (2023). A Survey on Optimization Techniques for Edge Artificial Intelligence (AI). *Sensors, 23* (3), 1-33. https://doi.org/10.3390/s23031279

[19]Alajlan, N. N., & Ibrahim, D. M. (2022). TinyML: Enabling of Inference Deep Learning Models on Ultra-Low-Power IoT Edge Devices for AI Applications. *Micromachines, 13* (6), 1-22. https://doi.org/10.3390/mi13060851

[20]Paes Salomon, C., Ferreira, C., Sant'Ana, W. C., Lambert-Torres, G., Borges Da Silva, L. E., Bonaldi, E. L., de Lacerda de Oliveira, L. E., Silva Torres, B. (2019). A Study of Fault Diagnosis Based on Electrical Signature Analysis for Synchronous Generators Predictive Maintenance in Bulk Electric Systems. *Energies, 12* (8), 1-16. https://doi.org/10.3390/en12081506

[21]Dilena, M., Fedele Dell'Oste, M., Fernández-Sáez, J., Morassi, A., Zaera, R. (2019). Mass detection in nanobeams from bending resonant frequency shifts. *Mechanical Systems and Signal Processing, 116*, 261–276. https://doi.org/10.1016/j.ymssp.2018.06.022

[22]Namdar, A. (2022). A robust principal component analysis-based approach for detection of a stator inter-turn fault in induction motors. *Protection and Control of Modern Power Systems, 7* (48), 1-24. https://doi.org/10.1186/s41601-022-00269-4

[23]Liu, Z., Zhang, P., He, S., Huang, J. (2021). A Review of Modeling and Diagnostic Techniques for Eccentricity Fault in Electric Machines. *Energies, 14* (14), 1-21. https://doi.org/10.3390/en14144296

[24]Brockmann, S., Schlippe, T. (2024). Optimizing Convolutional Neural Networks for Image Classification on Resource-Constrained Microcontroller Units. *Computers, 13* (7), 1-18. https://doi.org/10.3390/computers13070173

[25]Vela, D., Sharp, A., Zhang, R., Nguyen, T., Hoang, A., & Pianykh, O. S. (2022). Temporal quality degradation in AI models. Scientific Reports, 12(1), 11654. https://doi.org/10.1038/s41598-022-15245-z

[26]Mudraje, I., Vogelgesang, K., Herfet, T. (2025). A 1-D CNN inference engine for constrained platforms (Versión 1). *arXiv*. https://doi.org/10.48550/ARXIV.2501.17269

[27]Lamrini, M., Chkouri, M. Y., Touhafi, A. (2023). Evaluating the Performance of Pre-Trained Convolutional Neural Network for Audio Classification on Embedded Systems for Anomaly Detection in Smart Cities. *Sensors, 23* (13), 127. https://doi.org/10.3390/s23136227

[28]Ooko, S. O., Karume, S. M. (2024). Application of Tiny Machine Learning in Predicative Maintenance in Industries. *Journal of Computing Theories and Applications, 2* (1), 131–150. https://doi.org/10.62411/jcta.10929

[29]Bolat, Y., Murray, I., Ren, Y., Ferdosian, N. (2025). Decentralized Distributed Sequential Neural Networks Inference on Low-Power Microcontrollers in Wireless Sensor Networks: A Predictive Maintenance Case Study. *Sensors, 25* (15), 1-32. https://doi.org/10.3390/s25154595

[30]Azari, M. S., Flammini, F., Santini, S., Caporuscio, M. (2023). A Systematic Literature Review on Transfer Learning for Predictive Maintenance in Industry 4.0. *IEEE Access, 11*, 12887–12910. https://doi.org/10.1109/ACCESS.2023.3239784