

Dispositivo de ultra-bajo costo para la clasificación de licores usando tecnología TinyML

Ultra-Low-Cost Device for Liquor Classification using TinyML Technology

Ilber-Adonayt Ruge-Ruge¹   Ingrid-Carolina Ortiz-Álvarez¹  Fabián-R. Jiménez-López¹ 

¹ Grupo de Investigación I2E, Escuela de Ingeniería Electrónica, Facultad de Ingeniería, Universidad Pedagógica y Tecnológica de Colombia. Tunja - Boyacá, Colombia.

Resumen

Objetivo: Diseñar e implementar un sistema de identificación de licores mediante TinyML, empleando un sensor MQ-135 y un microcontrolador ESP32 de Espressif Systems. El objetivo es optimizar el procesamiento local para lograr alta precisión y baja latencia, validando así su viabilidad como solución accesible en entornos con recursos limitados.

Metodología: La metodología empleó una Red Neuronal Artificial (RNA) para clasificar licores según los compuestos volátiles detectados por el sensor. Se recopiló un conjunto de datos con 6000 mediciones y el modelo de RNA se entrenó en MATLAB® R2018a. Para la evaluación del rendimiento, el conjunto de datos se dividió en un 70 % para entrenamiento, un 15 % para validación y un 15 % para pruebas, utilizando el Error Cuadrático Medio (EMM) como métrica principal.

Resultados: El modelo de ANN implementado convergió tras 375 épocas, alcanzando un Error Cuadrático Medio (EMM) mínimo de $1,05 \times 10^{-4}$ y un coeficiente de correlación (R) de 1,0. Sintetizado en el microcontrolador ESP32, el modelo utilizó solo el 24,8 % de la memoria Flash y el 6,3 % de la RAM. El tiempo de inferencia registrado para la ejecución fue de 2,43 ms, lo que arroja una precisión de clasificación del 84,4 %. Esta solución rentable, con un precio de 15 USD, supera las opciones comerciales existentes.

Conclusiones: Alcanza un tiempo de inferencia excepcionalmente rápido de 2,43 ms con un consumo energético operativo mínimo, de entre 5 mA y 10 mA. Este rendimiento, sumado al bajo coste de fabricación (15 USD), contrasta marcadamente con las soluciones comerciales, que a menudo superan los cientos de dólares. En consecuencia, el sistema es muy viable para su implementación en aplicaciones integradas de control de calidad y de la industria alimentaria.

Palabras clave: TinyML, Clasificación de licores, Redes neuronales artificiales, sensor MQ135, Microcontrolador ESP32.

Abstract

Objective: To design and implement a liquor identification system using TinyML, employing an MQ-135 sensor and an ESP32 microcontroller from Espressif Systems. The objective is to optimize local processing for high precision and low latency, thereby validating its viability as an accessible solution in resource-constrained environments.

Methodology: The methodology employed an Artificial Neural Network (ANN) to classify liquors based on volatile compounds detected by the sensor. A dataset comprising 6,000 measurements was collected, and the ANN model was trained in MATLAB® R2018a. For performance evaluation, the dataset was split into 70% for training, 15% for validation, and 15% for testing, with the Mean Squared Error (MSE) used as the primary metric.

Results: The deployed ANN model converged after 375 epochs, achieving a minimal Mean Squared Error (MSE) of 1.05×10^{-4} and a correlation coefficient (R) of 1.0. Synthesized onto the ESP32 microcontroller, the model utilized only 24.8% of Flash memory and 6.3% of RAM. The inference time recorded for execution was 2.43 ms, yielding a classification accuracy of 84.4%. This cost-efficient solution, priced at 15 USD, outperforms existing commercial options.

Conclusions: It achieves an exceptionally fast inference time of 2.43 ms with a minimal operational energy consumption ranging from 5 mA to 10 mA. This performance, coupled with the low fabrication cost (15 USD), contrasts sharply with commercial solutions often exceeding hundreds of dollars. Consequently, the system is highly viable for implementation in embedded quality control and food industry applications.

Keywords: TinyML, Liquor classification, Artificial neural networks, MQ-135 sensor, ESP32 microcontroller.

¿Cómo citar?

Ruge-Ruge IA, Ortiz-Álvarez IC, Jiménez-Lopez FR. Dispositivo de ultra-bajo costo para la clasificación de licores usando tecnología TinyML. Ingeniería y Competitividad, 2025, 28(1)e-20215038

<https://doi.org/10.25100/iyv.v28i1.15038>

Recibido: 15/06/25

Revisado: 25/09/25

Aceptado: 21/01/26

Online: 10/02/26

Correspondencia

ilber.ruge@uptc.edu.co



¿Por qué se realizó este trabajo?

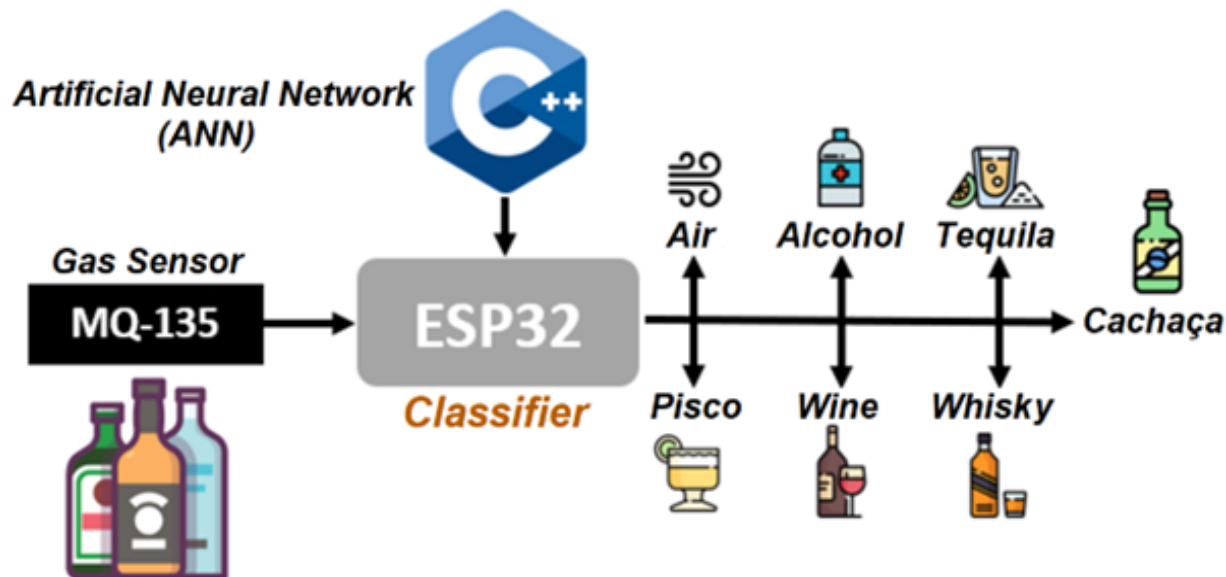
Este trabajo se realizó para demostrar la capacidad de Tiny Machine Learning (TinyML) para la clasificación de bebidas alcohólicas. TinyML permite la ejecución de modelos de aprendizaje automático en dispositivos con recursos limitados, lo que representa una solución innovadora en comparación con los enfoques convencionales. La tecnología se aplicó a la clasificación de seis tipos de bebidas alcohólicas: alcohol etílico medicinal, vino, pisco, cachaza, tequila y whisky. El objetivo principal fue desarrollar un sistema caracterizado por su bajo costo (aproximadamente 15 USD) y alta eficiencia computacional (tiempos de inferencia de 2,48 ms), lo que lo hace viable para su implementación en dispositivos embebidos.

¿Cuáles fueron los resultados más relevantes?

Los resultados indicaron una precisión de clasificación del 84 %, comparable a la de algunas soluciones comerciales más costosas, como el Gasboard-3210Plus (800 USD) o los analizadores Anton Paar (1500-2100 USD). Con una latencia de 32 ms, el sistema permite una clasificación rápida y eficiente en tiempo real.

¿Qué aportan estos resultados?

El sistema de clasificación desarrollado, con un costo estimado de tan solo 15 USD, representa una alternativa accesible para la autenticación de bebidas. Estos hallazgos sugieren su gran potencial de aplicación en el control de calidad, la detección de adulteraciones y la monitorización en la industria de bebidas alcohólicas.



Introducción

Actualmente, el desarrollo de dispositivos de bajo coste para la clasificación y detección de sustancias ha cobrado una relevancia significativa, especialmente en sectores como la industria alimentaria y de bebidas. Tiny Machine Learning (TinyML), que permite la ejecución de modelos de aprendizaje automático en dispositivos con recursos limitados, se presenta como una solución innovadora para tareas de clasificación en tiempo real (1,2).

Esta tecnología ofrece un potencial considerable para implementar sistemas eficientes y accesibles que faciliten la identificación de productos, como en el caso de los licores, sin necesidad de equipos caros o especializados. El factor determinante para seleccionar TinyML como tecnología central en este trabajo es su capacidad, como demuestran Schizas et al. (3), para mejorar el rendimiento y la eficiencia procesando datos localmente con latencia mínima y un consumo energético significativamente reducido. Al minimizar la transmisión de datos a la nube, TinyML refuerza la seguridad y reduce los costes operativos, consolidando así su papel como base para sistemas embebidos inteligentes, de bajo coste y altamente autónomos.

Se realizó una revisión sistemática de la literatura siguiendo la metodología PRISMA (*Preferred Reporting Items for Systematic Reviews and Meta-Analyses*) para identificar aplicaciones de TinyML en sistemas de clasificación. En la fase de identificación, las palabras clave “TinyML”, “Classification Machine Learning” y “Arduino” se emplearon en bases de datos disponibles en la Universidad Pedagógica y Tecnológica de Colombia (UPTC, Tunja): ScienceDirect (25), Web of Science (18), Scopus (30), Elsevier (12) y Google Scholar (40), dando 125 registros. Durante la fase de selección, se consideraron los títulos y resúmenes pertinentes y, posteriormente, en la fase de elegibilidad, solo se analizaron los artículos con texto completo disponible. En total, 12 estudios cumplieron los criterios de inclusión e incorporaron al análisis final.

En la literatura se han explorado diversas aplicaciones de TinyML e instrumentación de bajo coste. Tsoukas et al. (4) desarrollaron un sistema autónomo basado en TinyML para la detección de fugas de amoníaco y humo, logrando una puntuación F1 de 0,77 para humo y 0,70 para amoníaco. Bagheri et al. (5) evaluaron TinyML sobre microcontroladores y sensores ambientales para el monitoreo en tiempo real de contaminantes acuáticos, destacando su bajo coste y autonomía independiente de la nube. Atanane et al. (6) implementaron un sistema de detección de fugas de agua en edificios utilizando sensores acústicos, acelerómetros, Redes Neuronales Convolucionales (CNN) y TinyML en un Arduino Nano 33 BLE, logrando una precisión del 97,45%.

En cuanto a la clasificación de licores, Zhou et al. (7) informaron de una precisión del 100% en la clasificación de ocho marcas de licor chino utilizando sensores de gas y Análisis de Función Discriminante (DFA). Zhang et al. (8) utilizaron una nariz electrónica con sensores de ZnO dopados y técnicas de Análisis de Componentes Principales (PCA), Análisis de Clústeres (CA) y Cuantización de Vectores de Aprendizaje (LVQ), logrando hasta un 94,1% de precisión en la caracterización de vinagres chinos. Scorsone et al. (9) desarrollaron un morro electrónico utilizando sensores de polímero conductor (CP) para la detección temprana de incendios, reduciendo falsas alarmas mediante análisis PCA. Investigaciones relacionadas incluyen a Zhang et al. (10), que emplearon

sensores de gas de óxido de zinc (ZnO) dopados con MnO₂, TiO₂ y CO₂O₃, alcanzando una precisión del 89,3% con LVQ.

Además, Wongchoosuk et al. (11) diseñaron una nariz electrónica portátil con sensores híbridos de nanotubos de carbono–SnO₂ para detectar contaminación por metanol en whisky, identificando con éxito la contaminación por metanol al 1%. Otras aplicaciones relevantes incluyen el sistema inalámbrico de monitorización de monóxido de carbono (sensores MQ-9 y LPWAN) de Vega-Luna et al. (12), que alcanzó una cobertura de 11,8 km, y el sistema de Nagy et al. (13) para medir siete gases contaminantes usando sensores MQ-2 y MQ-5 en una Raspberry® Pi 3.

Finalmente, Jiang et al. (14) aplicaron modelos de Aprendizaje Automático, específicamente el Bosque Aleatorio, logrando una precisión superior al 90% en la predicción de la calidad del vino blanco. Específicamente, en el contexto colombiano, Botero-Valencia et al. (15) aplicaron con éxito TinyML en una estación de monitoreo ambiental, logrando una mejora del 2,67% en la precisión. Esto demostró un caso práctico de éxito de utilizar los microcontroladores de bajo recurso especificados para la detección ambiental de alto impacto en entornos locales, justificando aún más el enfoque arquitectónico adoptado en el presente estudio.

Los resultados relevantes de la revisión bibliográfica, que destacan el año, la solicitud y la ubicación, se muestran en la Tabla 1.

Tabla 1. Resultados de la revisión bibliográfica

Año	Autores	Aplicación	Microcontrolador	Variables medidas	Sensores utilizados	País
2023	Tsoukas, V. y otros. (4)	Detección de fugas de gas	No especificado	Humo, amoníaco	Sensores de gas	Grecia
2023	Bagheri, M. y otros. (5)	Monitorización ambiental en el agua	No especificado	Contaminantes del agua	Sensores ambientales	Irán
2023	Atanane, O. et al. Evaluaciones (6)	Detección de fugas de agua	Arduino™ Nano 33 BLE	Vibraciones, acústica	Acelerómetros, sensores acústicos	Marruecos
2011	Zhou, Q. et al. (7)	Clasificación de licores chinos	No especificado	Compuestos volátiles	Sensor MOX	China
2006	Zhang, Q. y otros. Evaluaciones (8)	Clasificación de vinagres chinos	No especificado	Compuestos volátiles	Sensores de ZnO dopados	China
2006	Scorsone, E. et al. Evaluaciones (9)	Detección temprana de incendios	No especificado	Humo	Sensores de polímero conductores	Italia
2005	Zhang, Q. y otros. Evaluaciones (10)	Clasificación de licores chinos	No especificado	Compuestos volátiles	Sensores de ZnO dopados	China
2010	Wongchoosuk, C. et al. (11)	Detección de metanol en whisky	No especificado	Metanol	Sensores CNT-SnO ₂	Tailandia

Año	Autores	Aplicación	Microcontrolador	Variables medidas	Sensores utilizados	País
2017	Vega-Luna, J. I. et al. (12)	Monitorización de monóxido de carbono	PIC18F8722	Monóxido de carbono	Sensor MQ-9	México
2020	Nagy, A. S. et al. (13)	Medición simultánea de gases	Raspberry® Pi 3	Gases, temperatura, humedad	MQ-2, MQ-5, DHT11	Cuba
2023	Jiang, X. y otros. (14)	Predicción de la calidad del vino	No aplicable	Propiedades fisicoquímicas	No aplicable	China
2023	Botero-Valencia, J. S. et al. (15)	Monitorización de la contaminación ambiental	ESP32-CAM	Aire, ruido, luz	SPS30, SCD30, SCD40, BME680, BH1750, AS7341	Colombia

La investigación complementaria ha explorado la aplicación de TinyML en diversas áreas distintas. Hayajneh et al. (16) propusieron un marco basado en TinyML y aprendizaje por transferencia para predecir la humedad del suelo en la agricultura inteligente, utilizando drones y sensores IoT. Karras et al. (17) desarrollaron algoritmos TinyML para la gestión de datos en IoT, mejorando la limpieza, compresión y almacenamiento de datos, con implementación en las plataformas Raspberry® Pi.

Schizas et al. (3) realizaron una revisión sistemática sobre TinyML en despliegues de IoT de bajo consumo, destacando beneficios en cuanto a transición, ancho de banda, seguridad, privacidad, latencia, eficiencia energética y bajo coste. Dutta y Bharali (18) analizaron la integración de TinyML en IoT, destacando ventajas como la reducción de costes y la independencia de la nube, junto con los retos de hardware asociados. Srinivasagan et al. (19) diseñaron un sensor TinyML para estimar la vida útil de los dátiles frescos utilizando espectrometría VisNIR.

Capogrosso et al. (20) clasificaron los enfoques de desarrollo en tres categorías: algorítmicos, centrados en hardware y co-diseño. Banbury et al. (21) propusieron referencias para evaluar las plataformas TinyML. Hymel et al. (22) presentaron Edge Impulse, una plataforma que simplifica el desarrollo y despliegue de modelos. Finalmente, Lê et al. (23) revisaron técnicas de optimización en redes neuronales para microcontroladores con recursos limitados. Las aplicaciones relevantes que utilizan la tecnología TinyML se muestran en la Tabla 2.

Tabla 2. Resultados de la revisión bibliográfica sobre aplicaciones de TinyML.

Año	Autores	Aplicación	País
2023	Hayajneh, A. M. et al. (16)	Predicción de la humedad del suelo en la agricultura inteligente. UAVs, dispositivos IoT TinyML	Jordania / Reino Unido
2024	Karras, A. et al. (17)	Gestión de Big Data en IoT con algoritmos TinyML	Grecia
2022	Schizas, N. et al. (3)	TinyML en despliegues de IoT de bajo consumo	Grecia
2021	Dutta, L. y Bharali, S. (18)	Integración de TinyML en IoT	India
2023	Srinivasagan, R. et al. (19)	Estimación de la vida útil de los dátiles frescos	Arabia Saudí / Egipto
2023	Capogrosso, L. et al. (20)	Revisión de los algoritmos de aprendizaje de TinyML	Italia
2020	Banbury, C. R. et al. (21)	Benchmarking de plataformas TinyML	No especificado
2023	Hymel, S. et al. (22)	Plataforma Edge Impulse para el despliegue de TinyML	No especificado
2023	Lê, M. T. et al. (23)	Técnicas de optimización para redes neuronales en microcontroladores para TinyML	Francia

Con este contexto, este estudio pretende proponer un dispositivo basado en TinyML de ultra bajo costo con el objetivo de clasificar las bebidas alcohólicas. Las bebidas alcohólicas son un contexto relevante donde la rapidez y la precisión son fundamentales, ya que es importante hacer la clasificación correcta dentro del orden de los términos. El objetivo es crear un camino preciso, portátil y fácil de seguir para clasificar los tipos de bebidas alcohólicas basándose en sensores y algoritmos convencionales. El presente trabajo aborda la brecha identificada proponiendo un dispositivo de ultra bajo coste basado en la tecnología TinyML para la clasificación de licores, una aplicación donde tanto la precisión como la rapidez son factores críticos.

El término ultra-bajo coste caracteriza sistemas embebidos cuyo coste total de materiales (Lista de Materiales – BOM) es mínimo, priorizando componentes económicos sin comprometer la funcionalidad esencial del sistema. La solución está implementada en el microcontrolador ESP32 de Espressif Systems, seleccionado específicamente debido a su bajo coste de mercado (aproximadamente 8 USD), la amplia disponibilidad en el mercado y la extensa documentación para desarrolladores. Este microcontrolador cuenta con un procesador Tensilica LX6 de doble núcleo (hasta 240 MHz), 520 KB de SRAM, 4 MB de memoria Flash e interfaces UART, SPI, I2C y ADC de 12 bits.

Esta arquitectura permite la ejecución eficiente de modelos de aprendizaje automático a pesar de sus limitados recursos computacionales, integrando sensores de bajo coste para desarrollar una

herramienta precisa y económicamente accesible para la identificación de bebidas alcohólicas. Se prevé que este sistema propuesto contribuya tanto a la eficiencia operativa industrial como a la democratización de tecnologías avanzadas en contextos con restricciones presupuestarias (24,25).

Metodología

El sistema de clasificación se basa en el sensor de gas MQ-135 y en una Red Neuronal Artificial (ANN). El sensor MQ-135 es un dispositivo semiconductor de óxido metálico (MOS) ampliamente adoptado capaz de detectar diversos gases, incluyendo monóxido de carbono (CO), alcohol, dióxido de carbono (CO₂), tolueno (C₆H₆), amoníaco (NH₃), alcohol y dióxido de carbono (26). El principio de funcionamiento del sensor se basa en la variación de la resistencia del material sensible R_s en respuesta a la concentración de gas.

La calibración se realiza siguiendo las recomendaciones del fabricante para determinar el coeficiente R_0 , que corresponde a la resistencia base del sensor en aire limpio. Este valor R_0 es crucial para normalizar la respuesta del sensor, ya que la relación R_s / R_0 aísla el cambio de resistencia específicamente debido a la presencia de concentraciones de gas objetivo. Esta normalización es esencial para validar la robustez del modelo frente a fluctuaciones ambientales típicas, como cambios de temperatura y humedad. La relación entre la resistencia medida R_s y la concentración de gas C se expresa mediante un modelo logarítmico derivado de la curva característica del sensor, tal como se presenta en la Ecuación 1.

$$\frac{R_s}{R_0} = A * (C)^{-B} \quad (1)$$

donde R_s es la resistencia del sensor a una concentración de gas C , R_0 es la resistencia en aire limpio y A con B son constantes determinadas experimentalmente específicas para el tipo de gas.

Los RNA se emplearon para identificar licores basándose en sus firmas distintivas de compuestos orgánicos volátiles (COV), lo que resultó en modelos robustos y de alta precisión adecuados para el control de calidad y la autenticación de bebidas. Específicamente, el sistema está diseñado para realizar control de calidad verificando la firma de COV frente a un estándar conocido y la autenticación de la bebida detectando variaciones no autorizadas, como dilución o la presencia de adulterantes como el metanol. En términos generales, una red de perceptrón multicapa (MLP) puede representarse mediante las siguientes expresiones mostradas en las Ecuaciones 2 y 3.

$$z_j = \sum_{i=1}^n w_{ij} x_i + b_j \quad (2)$$

$$y_k = f(z_j) \quad (3)$$

donde x_i representa la variable de entrada correspondiente al valor de voltaje obtenido del sensor MQ-135, w_{ij} son los pesos sinápticos, b_j son los sesgos asociados y y_k es la función de activación (27).

El entrenamiento en redes neuronales se llevó a cabo en MATLAB® R2018a utilizando el 'nntraintool', que simplifica la configuración y el entrenamiento mediante algoritmos como la retropropagación y proporciona herramientas gráficas para evaluar el rendimiento del modelo

(28). El modelo optimizado se implementó entonces en una placa ESP32, un microcontrolador con un rendimiento adecuado para aplicaciones de IoT, TinyML y procesamiento de bordes. El procesamiento en el borde se entiende como la capacidad de ejecutar tareas de análisis y toma de decisiones directamente en el dispositivo sin depender de una conexión continua en la nube (29).

La placa de desarrollo ESP32 integra un procesador Tensilica Xtensa LX6 de doble núcleo que opera hasta 240 MHz, 520 KB de SRAM y 4 MB\$ de memoria Flash, junto con interfaces de comunicación como Interfaz Periférica Serial (SPI), Circuito Interintegrado (I2C), Receptor-Transmisor Universal Asíncrono (UART) y un Convertidor Analógico-Digital (ADC) de 12 bits. Esta arquitectura permite una integración eficiente con sensores y módulos de adquisición de datos (30). La Figura 1 muestra el diagrama de bloques del sistema TinyML propuesto para la clasificación de licores.

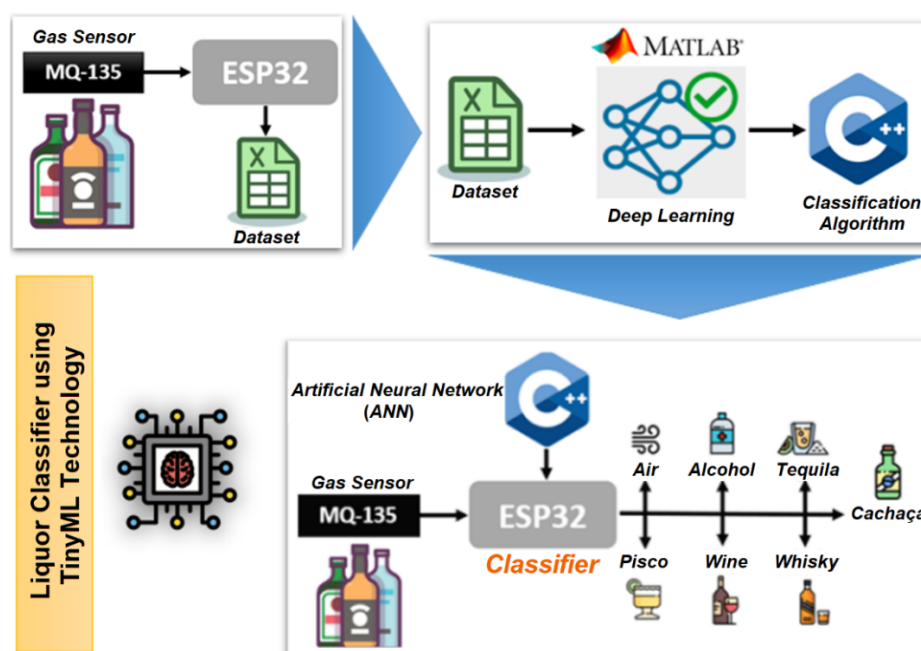


Figura 1. El flujo de trabajo de desarrollo del clasificador de licor TinyML.

La conexión con el sensor MQ-135 se gestionaba a través de la biblioteca 'MQUnifiedsensor.h' en el IDE estándar de Arduino™, permitiendo la lectura de valores y la configuración de parámetros para cada gas medido.

El conjunto de datos para entrenar la red neuronal se construyó a partir de mediciones del sensor MQ-135, capturando la concentración de compuestos volátiles como alcohol, hexano, monóxido de carbono (CO), benceno y gas licuado de petróleo (GLP). El proceso de adquisición implicó muestras de aire ambiente, alcohol etílico medicinal, vino, pisco, cachaça, tequila y whisky. Se registraban los valores de voltaje de referencia y el coeficiente R0 para monitorizar la estabilidad del sensor.

El sistema final de clasificación se diseñó con siete categorías de salida: Alcohol, Vino, Pisco, Cachaça, Tequila, Whisky y Aire Ambiente (usado como referencia). Para asegurar una representación robusta de los datos, se recogieron 100 muestras para cada una de las siete

categorías, con 10 repeticiones para cada muestra para tener en cuenta la variación de captura de señales. Esto generó un total inicial de 7 categorías ×: 100 muestras × , 10 repeticiones, resultando en 7.000 registros. Como se señaló en el análisis preliminar, las primeras 50 lecturas iniciales fueron descartadas debido al tiempo de sedimentación y a la inestabilidad inherente del sensor MQ-135 durante el calentamiento. En consecuencia, el conjunto de datos utilizable final consistió en 6.950 registros. Las características de entrada capturaban la concentración de los distintos compuestos volátiles, y la salida se etiquetaba usando un esquema binario correspondiente a la presencia de cada tipo de licor o aire ambiente.

La Figura 2 muestra un ejemplo de una parte de la estructura del conjunto de datos dentro de Excel™, con cada captura y la clasificación correspondiente en filas.

Alcohol	Hezane	CO	Benzene	LPG	Volt	R0	Alcohol	Wine	Pisco	Cachaca	Tequila	Whisky	Air
0.13	878.1	28949.11	0.64	3840.68	4.35	0	1	0	0	0	0	0	0
0.13	888.46	28498.36	0.64	3840.68	4.35	0	1	0	0	0	0	0	0
0.13	888.46	28949.11	0.64	3840.68	4.35	0	1	0	0	0	0	0	0
0.07	324.93	7630.96	0.25	1250.89	4.11	0.01	0	0	1	0	0	0	0
0.07	327.95	7726.26	0.25	1237.77	4.11	0.01	0	0	1	0	0	0	0
0.08	331.01	7822.94	0.25	1264.17	4.12	0.01	0	0	1	0	0	0	0
0.05	175.79	3347.78	0.14	609.18	3.94	0.01	0	0	0	0	1	0	0
0.05	175.79	3347.78	0.14	614.84	3.94	0.01	0	0	0	0	1	0	0
0.05	174.37	3311.51	0.14	609.18	3.94	0.01	0	0	0	0	1	0	0
0	0.07	0.09	0	0.08	0.92	0.15	0	0	0	0	0	0	1
0	0.07	0.09	0	0.08	0.92	0.15	0	0	0	0	0	0	1
0	0.07	0.09	0	0.08	0.92	0.15	0	0	0	0	0	0	1

Figura 2. Estructura del conjunto de datos para entrenar el sistema de clasificación de licor TinyML.

El entrenamiento de redes neuronales se llevó a cabo en MATLAB® R2018a utilizando el ‘nntraintool’, facilitando la configuración y el entrenamiento mediante el algoritmo Levenberg–Marquardt (31). Este algoritmo aprovecha una combinación del descenso más pronunciado y los métodos de Gauss–Newton para minimizar eficientemente el Error Cuadrático Medio (MSE). Matemáticamente, la actualización de peso se define como se muestra en la Ecuación 4.

$$\Delta w = -[J^T J + \mu I]^{-1} J^T e$$

(4)

donde **J** es la matriz jacobiana de las derivadas parciales de error respecto a los pesos, **e** es el vector de error, **I** es la matriz identidad y μ es un factor de ajuste que regula la transición entre el comportamiento de descenso más pronunciado (cuando μ es grande) y el método de Gauss–Newton (cuando μ es pequeño).

La arquitectura de la red estaba configurada con 5 neuronas de entrada, una capa oculta y 6 neuronas de salida, correspondientes a las clases de clasificación finales. Se realizaron pruebas variando el número de neuronas en la capa oculta para analizar su efecto sobre la convergencia. La

función principal de pérdida utilizada fue el Error Cuadrático Medio (MSE), y los datos se dividieron en 70% para entrenamiento, 15% para validación y 15% para pruebas.

Resultados y discusión

Preprocesamiento y estabilidad de datos de sensores

Los datos utilizados para el entrenamiento excluyen las primeras 50 lecturas del proceso de muestreo para cada clase de entrada, ya que se observó que los valores del sensor permanecían inestables durante ese intervalo de tiempo, como se muestra en la Figura 3. Esta variabilidad inicial, que se estabiliza tras el periodo de calentamiento, obliga a excluir muestras tempranas para asegurar que el modelo se entrene únicamente con datos fiables, maximizando así la precisión predictiva del modelo.

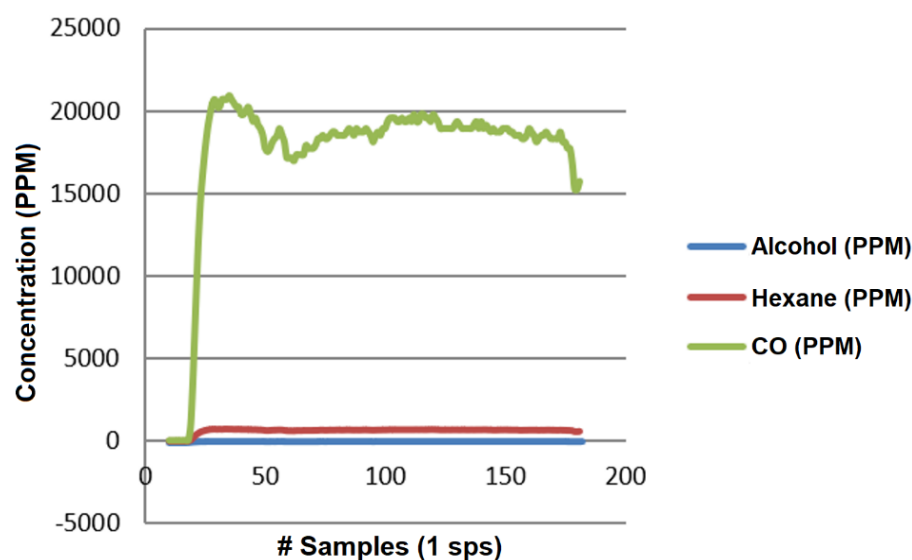


Figura 3. Variabilidad del sensor MQ135 durante una nueva medición.

En cuanto al preprocesamiento, este paso fue fundamental para asegurar que los datos de entrenamiento representaran a un sensor comportándose en estado estacionario, y no permitía al modelo aprender los patrones de inestabilidad circundantes, sino las características compuestas de las sustancias objetivo. Implementar esta forma de condicionamiento de datos es una práctica común en aplicaciones de aprendizaje automático basadas en sensores para ayudar en mejoras de rendimiento y fiabilidad.

Rendimiento y convergencia de redes neuronales

El proceso de entrenamiento de redes neuronales demostró una convergencia adecuada bajo condiciones de estabilidad del gradiente y una variación mínima entre los conjuntos de entrenamiento, validación y prueba. Esta estabilidad se evidencia con la reducción progresiva del Error Cuadrático Medio (MSE) durante 375 épocas, alcanzando un valor mínimo de $2,8083 \times 10^{-9}$ en la fase de validación (véase la Figura 4). Además, el gradiente final ($9,91 \times 10^{-8}$) y el error ($1,05 \times 10^{-9}$) se mantuvieron dentro de rangos óptimos de convergencia, sin mostrar signos discernibles de sobreajuste durante el proceso de entrenamiento.

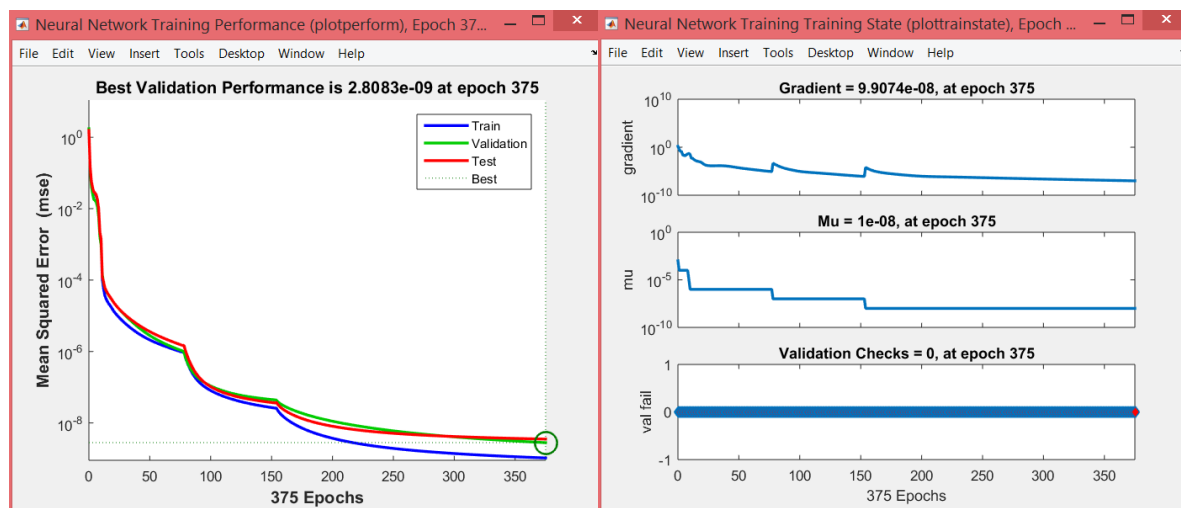


Figura 4. Rendimiento de entrenamiento según el modelo de clasificación.

Análisis de regresión de modelos

Además de la reducción del MSE durante el entrenamiento, la evaluación de regresión entre los resultados predichos y los valores objetivo mostró un ajuste del modelo extremadamente alto. Como se ilustra en la Figura 5, los coeficientes de correlación (R) para los conjuntos de entrenamiento, validación, prueba y datos totales alcanzaron valores iguales a 1,0. Este resultado sugiere una relación lineal casi perfecta entre las salidas de la red y los valores objetivo esperados

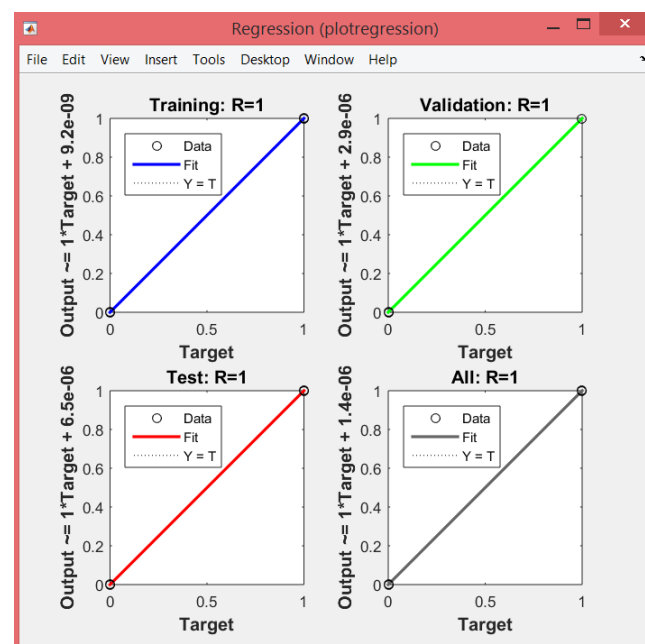


Figura 5. Evaluación regresiva del modelo de clasificación.

La observación de valores MSE que se acercan a cero y $R = 1$ en todas las particiones de datos, a pesar del tamaño relativamente limitado del conjunto de datos, se atribuye principalmente a dos factores: (1) la separabilidad inherente de las clases y (2) la eficiencia del algoritmo de optimización de Levenberg–Marquardt. Dado que los perfiles de compuestos orgánicos volátiles (COV) de las siete categorías (seis licores y aire ambiente) son químicamente distintos, la red neuronal pudo aprender límites altamente separables con un error mínimo.

La naturaleza no lineal de la respuesta del sensor, cuando se mapea a través de la capa oculta del MLP, probablemente da lugar a un espacio de proyección claro, lo que lleva a un ajuste casi perfecto en los conjuntos de entrenamiento y validación. Este comportamiento también ha sido reportado en estudios similares, como el de Aljohani et al. (32), quienes obtuvieron $R = 1$ al aplicar redes neuronales entrenadas con el algoritmo de Levenberg–Marquardt para analizar recubrimientos de alambre en fluidos de Sisko, utilizando tamaños y épocas de conjuntos de datos comparables. No obstante, considerando el tamaño limitado del presente conjunto de datos, este resultado se interpreta como una alta capacidad de ajuste de patrones bajo condiciones controladas, más que como una validación definitiva de la capacidad total de generalización del modelo a muestras desconocidas o adulteradas.

Despliegue de modelos y utilización de recursos

El 'nntraintool' generó una representación matemática del modelo entrenado (a través de la opción 'desplegar solución –> Función Matlab'), que posteriormente se ajustó al lenguaje C/C++ para su uso en el IDE Arduino™ y la implementación final en el microcontrolador ESP32. Este proceso de ajuste incluía la definición sintáctica de los vectores de pesos y las funciones de normalización de entrada/salida. Las Figuras 6 y 7 presentan los extractos más relevantes del código desarrollado en la versión 1.8.19 del IDE de Arduino™, ilustrando estos ajustes.

```

/*****Globals E-Nose*****/
//double x_Input[7]={0.05,180.14,3459.37,0.14,632.2,3.95,0.01};
double x_Input[7];
double x_Offset[] = {0,0.06,0.08,0,0.07,0.9,0};
double x_Ganancia[]={15.3846153846154,0.00225123818099955,0.0000680081732222579,3.125,0.0005138:
double x_Y_min= -1;
double a1[20];
double a2[7];
double y_Offset[] = {0,0,0,0,0,0,0};
double y_Ganancia[]={2,2,2,2,2,2,2};
double y_Y_min= -1;

// Layer 1
double b1[] = {-1.6523867461356561,6.1138055577454509,2.5944933158635131,0.82639102310003743,2.4:
double IW1_1[20][7] = {{0.75440637200062111,0.66256452398938537,1.5879691272379033,0.95120615507:

//Layer 2
double b2[] = {0.79132620384296259,-0.97149375244564684,-0.098346155977758351,-0.941493427197531
double LW2_1[7][20] = {{1.3472180990584768,0.024523386294063212,-0.50440969930700508,0.03169865:

```

Figura 6. Ajuste sintáctico de vectores de peso desde Matlab Function a C/C++.

```

//***** Layer 1 a1 = (b1+(IW1_1*Data_Input)) *****
//***** TinyML_Model Liquor identifier *****
//*****
void evaluate_TinyML_Model()
{
    x_Input[0]= coPPM;
    x_Input[1]= alcoholPPM;
    x_Input[2]= co2PPM;
    x_Input[3]= toluenPPM;
    x_Input[4]= nh4PPM;
    x_Input[5]= sensor_volt;
    x_Input[6]= R0;

    //***** Data Input Normalized [-1,1] *****
    for(int i=0;i<7;i++)
    {
        x_Input[i]= x_Input[i]-x_Offset[i];
        x_Input[i]= x_Input[i]*x_Ganancia[i];
        x_Input[i]= x_Input[i]+x_Y_min;
    }

    //***** Layer 1 a1 = (b1+(IW1_1*Data_Input)) *****
    for(int i=0;i<20;i++)
    {
        double Acum=0.0;
        for(int j=0;j<7;j++)
        {
            Acum=IW1_1[i][j]*x_Input[j]+Acum;
        }
        a1[i]=Acum+b1[i];
        a1[i] = 2/(1+exp(-2*a1[i]))-1;
    }

    //***** Layer 2 a2 = (b2+(LW2_1*a1)) *****
    for(int i=0;i<7;i++)
    {
        double Acum=0.0;
        for(int j=0;j<20;j++)
        {
            Acum=LW2_1[i][j]*a1[j]+Acum;
        }
        a2[i]= Acum+b2[i];
        a2[i]= a2[i]-y_Y_min;
        a2[i]= a2[i]/y_Ganancia[i];
        a2[i]= a2[i]+y_Offset[i]; //**** Data Output Reverse Normalized [-1,1]
    }
}

```

Figura 7. Ajuste sintáctico del modelo de clasificación de Matlab Function a C/C++.

El proceso de cargar el modelo en el ESP32 demuestra un uso eficiente de los recursos del sistema embebido. La memoria Flash utilizada medía 297.525, representando el 24,8% de la memoria Flash disponible (1,2 MB para la aplicación de partición de aplicaciones), lo que deja margen suficiente para futuras expansiones. En cuanto al uso de RAM, sumaba 20.432 bytes, equivalente al 6,3% de los 327.680 bytes de RAM disponibles, lo que indica una implementación ligera con bajo impacto en los recursos del microcontrolador. La latencia de programación era baja, con 2,3 segundos necesarios para escribir, lo que indicaba una subida rápida y estable del firmware, optimizando la ejecución del modelo en un entorno embebido.

El tiempo estimado de ejecución del algoritmo en el ESP32 es de aproximadamente 2,48 ms. Este cálculo considera que el código cargado ocupa 297.525 bytes en memoria Flash y que cada instrucción, con un tamaño medio de 2 bytes, requiere 2 ciclos de reloj para su ejecución. Operando a una frecuencia de 240 MHz, el microcontrolador procesa 595.050 ciclos, lo que refleja una latencia extremadamente baja y una ejecución en tiempo real eficiente, adecuada para aplicaciones de clasificación embebida y reconocimiento de patrones.

Los recursos utilizados en el despliegue del modelo TinyML en el ESP32 se resumen en la Tabla 3. La velocidad de ejecución alcanzable y el bajo uso de recursos indican que TinyML realmente ofrece una capacidad para aprendizaje automático avanzado en hardware económico y con recursos limitados; un gran beneficio por implicar las decisiones en tiempo real que a menudo se requieren en la producción de alimentos y bebidas.

Tabla 3. Resumen de los recursos utilizados en el despliegue del modelo TinyML en el ESP32.

Tipo de recurso	Valor usado	Descripción
Memoria Flash	297525 bytes (24,8%)	El espacio ocupado por el código en la memoria Flash.
Memoria RAM	20432 bytes (6,3%)	RAM utilizada para ejecutar el modelo.
Latencia de programación	2,3 segundos	Tiempo que tarda en cargar el código en el ESP32.
Tiempo de ejecución	2,48 ms	Latencia (tiempo de procesamiento).

Validación y rendimiento del sistema

La validación del sistema de clasificación de licor se realizó en un entorno que garantizaba condiciones estables de temperatura (22°C) y humedad relativa (75%) para minimizar variaciones externas. Se seleccionaron seis tipos de licores para su prueba: alcohol etílico medicinal, vino, pisco, cachaça, tequila y whisky. Durante la fase de pruebas, el sistema embebido capturó datos en tiempo real, ejecutó el modelo de clasificación y mostró la predicción en el Terminal Virtual del IDE Arduino™ configurado a una velocidad de 9600.

Para evaluar la estabilidad del sistema, se realizaron 30 mediciones por muestra, acumulando un total de 180 pruebas, lo que permitió analizar la repetibilidad y consistencia de las predicciones. Las predicciones se compararon con los valores esperados, lo que proporcionó un porcentaje de precisión que refleja la capacidad del sistema para identificar correctamente cada tipo de licor. Los resultados indican una precisión media del 84,4%, con valores individuales que oscilan entre el 80,0% y el 86,7%, dependiendo del licor específico analizado. En total, de las 180 muestras evaluadas, el sistema clasificó correctamente 152, demostrando un rendimiento sólido en la identificación de las diferentes bebidas. Los resultados de validación del dispositivo de clasificación TinyML se muestran en la Tabla 4.

Tabla 4. Resultados de validación del dispositivo de clasificación TinyML.

Tipo de licor	Muestras totales	Correctamente clasificado	Precisión (%)	Partitura
Alcohol etílico	30	26	86.7	0.87
Vino	30	25	83.3	0.83
Pisco	30	26	86.7	0.86
Cachaça	30	25	83.3	0.82
Tequila	30	24	80.0	0.78
Whisky	30	26	86.7	0.88
Promedio general	180	152	84.4	0.84

Coste-efectividad y análisis comparativo

Por último, un sistema se considera de ultra bajo coste cuando prioriza la reducción de costes sin comprometer la funcionalidad ni la precisión requerida. Los criterios clave incluyen el coste total del hardware, la disponibilidad comercial de componentes, el bajo consumo energético, la facilidad de implementación y la escalabilidad. Según Ciuffoletti (33), los sistemas IoT de bajo coste logran un equilibrio entre rendimiento, simplicidad y sostenibilidad económica, lo que los hace adecuados para entornos educativos, comerciales o con recursos limitados.

En este contexto, el sistema desarrollado cumple estos criterios empleando hardware accesible—ESP32 (5–10 USD) y sensor MQ-135 (≈ 5 USD)—alcanzando un coste total de aproximadamente 15 USD. Este coste es significativamente menor que el de equipos comerciales como el Gasboard-3210Plus de Cubic Instruments (34) o el sistema de análisis de Anton Paar (35), que a menudo superan los 800 dólares estadounidenses.

La Figura 8 demuestra la aplicación física del dispositivo de clasificación de licor desarrollado basado en TinyML como compacto y práctico. La Tabla 5 describe una comparación directa de los costes estimados considerando el sistema TinyML propuesto y los sistemas comerciales discutidos, demostrando la gran rentabilidad del prototipo desarrollado, pero mostrando aún así un grado razonable de precisión para su uso previsto.

El dispositivo TinyML, gracias a esta rentabilidad y precisión establecida en la clasificación, puede considerarse una tecnología realista y democratizadora para la autenticación y control de calidad de bebidas en mercados o aplicaciones centradas en la restricción de costes. El equilibrio rendimiento/precio resulta atractivo para una adopción futura en industrias que quizá no tengan acceso a equipos de laboratorio de élite.

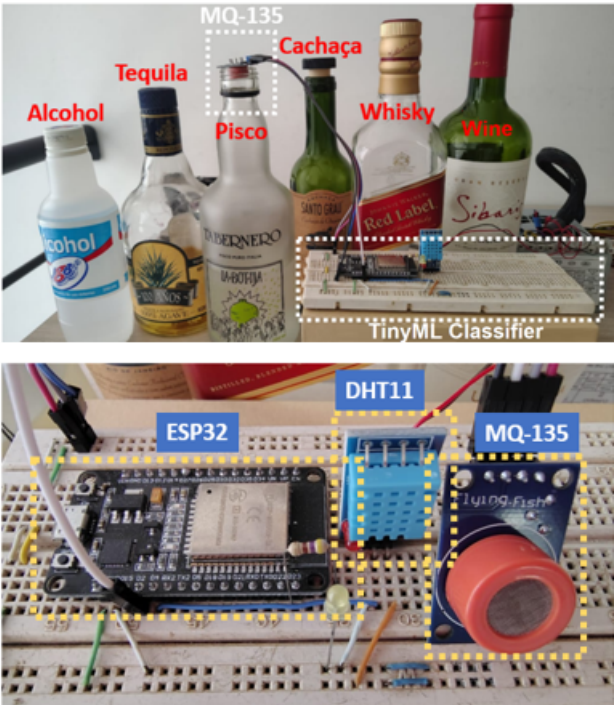


Figura 8. Dispositivo de clasificación de licor basado en TinyML.

Tabla 5. Comparación de costes entre el sistema de clasificación basado en TinyML y algunas soluciones comerciales.

Característica	Clasificador TinyML	Gasboard-3210Plus	Medidor de vides Anton Paar	Medidor de alcohol Anton Paar
Coste estimado	15 USD	800 USD	2100 USD	1500 USD
Hardware principal	Microcontrolador ESP32 + Sensor MQ135	Sensores de espectrometría de gases	Sensores electroquímicos	Sensores electroquímicos
Obligatorio Software	Código en C/C++ (IDE Arduino™)	Software propietario	Software propietario	Software propietario

Conclusiones

El sistema de clasificación de licores, basado en TinyML, desarrollado en este estudio, ha demostrado ser una solución eficiente y rentable cuando se compara con dispositivos comerciales. Con un coste estimado de hardware de aproximadamente 15 dólares estadounidenses, el prototipo desarrollado supera significativamente el acceso a alternativas costosas, como los analizadores Gasboard-3210Plus (800 dólares estadounidenses) o Anton Paar (1500–2100 dólares). Además,

el rápido tiempo de inferencia de 2,48 ms y la huella mínima de memoria (solo el 24,8% de la memoria Flash disponible utilizada) respaldan la viabilidad del sistema para su despliegue en sistemas embebidos con recursos limitados sin comprometer el rendimiento.

En términos de precisión predictiva, el clasificador alcanzó una tasa media de clasificación del 84,4%. La precisión varió según el tipo específico de licor, llegando hasta el 86,7% para la identificación de alcohol etílico medicinal, pisco y whisky. Aunque estos resultados son prometedores para un sistema que depende de hardware sencillo, aún existe margen para mejorar el rendimiento mediante el empleo de modelos de aprendizaje automático más robustos o la integración de sensores adicionales capaces de captar una gama más amplia de características químicas. No obstante, el nivel de rendimiento alcanzado es competitivo debido a la sencillez y el bajo coste de la plataforma de hardware utilizada.

Esta investigación sienta una base para futuras investigaciones sobre la aplicación de TinyML en la detección y clasificación de sustancias. La optimización de modelos, el uso de técnicas avanzadas de preprocesamiento y la adaptación del sistema a otros tipos de líquidos o gases podrían ampliar sustancialmente su aplicabilidad en industrias como el procesamiento de alimentos y el control de calidad. La combinación sinérgica de bajo coste, facilidad de implementación y tiempos de inferencia rápidos posiciona eficazmente esta tecnología como una alternativa viable y democratizadora en entornos con estrictas limitaciones presupuestarias o de recursos.

Declaración de contribución de autoría de créditos

Conceptualización – Ideas: Ilber Ruge, Ingrid Ortiz, Fabián Jiménez; Curación de datos: Ingrid Ortiz; Análisis formal: Ilber Ruge, Fabián Jiménez; Investigación: Ilber Ruge, Ingrid Ortiz, Fabián Jiménez; Metodología: Ilber Ruge; Supervisión: Ingrid Ortiz; Validación: Ilber Ruge, Ingrid Ortiz; Guion – Preparación: Fabián Jiménez; Escritura - Revisión y edición: Ilber Ruge; Preparación: Ilber Ruge, Ingrid Ortiz, Fabián Jiménez.

Implicaciones éticas: Los autores no tienen implicaciones éticas que deban declararse en la redacción y publicación de este artículo.

Conflictos de interés: Los autores no declaran.

Referencias

1. Saraan A, Hussain N, Zahara SM. Tiny Machine Learning (TinyML) Systems [Internet]. 2024 [cited in mar 2025]. Available in: https://www.researchgate.net/publication/386579238_Tiny_Machine_Learning_TinyML_Systems
2. Murshed MGS, Murphy C, Hou D, Khan N, Ananthanarayanan G, Hussain F. Machine learning at the network edge: A survey. ACM Comput Surv [Internet]. 2021 [cited in mar 2025]; 54(8):Article 170, 1-37. <https://doi.org/10.1145/3469029>
3. Schizas N, et al. TinyML for ultra-low power AI and large scale IoT deployments: A systematic review. 2022. <https://doi.org/10.3390/fi14120363>

4. Tsoukas V, Gkogkidis A, Boumpa E, Papafotikas S, Kakarountas A. A gas leakage detection device based on the technology of TinyML. En: 2023 12th International Conference on Modern Circuits and Systems Technologies (MOCAST); 2023. p. 1-4. IEEE. <https://doi.org/10.1109/MOCAST54814.2022.9837510>
5. Bagheri M, Farshforoush N, Bagheri K, Shemirani AI. Application of TinyML in water environments. *Environ Monit Assess*. 2023;195(5):1-14.
6. Atanane O, Mourhir A, Benamar N, Zennaro M. Smart buildings water leakage detection using TinyML. *Sensors*. 2023;23(9):4473. <https://doi.org/10.3390/s23229210>
7. Zhou Q, Zhang S, Li Y, Xie C, Li H, Ding X. A Chinese liquor classification method based on liquid evaporation and a metal oxide gas sensor. *Sens Actuators B Chem*. 2011;158(1):241-6. <https://doi.org/10.1016/j.snb.2011.08.015>
8. Zhang Q, Xie C, Zhang S, Wang A, Zhu B, Wang L, et al. Identification and pattern recognition analysis of Chinese liquors by doped nano ZnO gas sensor array. *Sens Actuators B Chem*. 2005;110(2):370-6. <https://doi.org/10.1016/j.snb.2005.02.017>
9. Scorsone E, Pisanelli AM, Persaud KC. Development of an electronic nose for fire detection. *Sens Actuators B Chem*. 2006;116(1-2):55-61. <https://doi.org/10.1016/j.snb.2005.12.059>
10. Zhang Q, Zhang S, Xie C, Zeng D, Fan C, Li D. Characterization of Chinese vinegars by electronic nose. *Sens Actuators B Chem*. 2006;120(2):694-9. <https://doi.org/10.1016/j.snb.2006.01.007>
11. Wongchoosuk C, Wisitsoraat A, Tuantranont A, Kerdcharoen T. Portable electronic nose based on carbon nanotube-SnO₂ gas sensors and its application for detection of methanol contamination in whiskeys. *Sens Actuators B Chem*. 2010;147(2):392-9. <https://doi.org/10.1016/j.snb.2010.03.072>
12. Vega-Luna JI, Lagos-Acosta MA, Salgado-Guzmán G. Monitoreo de concentración de monóxido de carbono usando tecnología long-range. In: Congreso Internacional sobre Innovación y Desarrollo Tecnológico (CIINDET); 2017. p. 1-6. IEEE. <https://doi.org/10.17163/ings.n18.2017.09>
13. Nagy AS, Polanco Risquet A, Martínez de la Cotería OL, Carralero Ibargollen O. Medición simultánea de gases con sensores MQ. *RIELAC Revista de Ingeniería Electrónica, Automática y Comunicaciones*. 2020;41(1):34-43. Disponible en: <http://scielo.sld.cu/pdf/eac/v41n1/1815-5928-eac-41-01-34.pdf>
14. Jiang X, Liu X, Wu Y, Yang D. White wine quality prediction and analysis with machine learning techniques. *Highlights in Science, Engineering and Technology*. 2023;39(1):321-326. doi: <https://doi.org/10.54097/hset.v39i.6548>
15. Botero-Valencia JS, Barrantes-Toro C, Márquez-Viloria D, Pearce JM. Low-cost air, noise, and light pollution measuring station with wireless communication and TinyML. *HardwareX*. 2023;14:e00422. <https://doi.org/10.1016/j.ohx.2023.e00477>
16. Hayajneh AM, et al. Tiny Machine Learning on the edge: A framework for transfer learning empowered UAV-assisted smart farming. 2023. <https://doi.org/10.1049/smc2.12072>
17. Karras A, et al. TinyML algorithms for big data management in large-scale IoT systems. 2024. <https://doi.org/10.3390/fi16020042>
18. Dutta L, Bharali S. TinyML meets IoT: A comprehensive survey. 2021. <https://doi.org/10.1016/j.iot.2021.100461>
19. Srinivasagan R, Mohammed M, Alzahrani A. TinyML-Sensor for shelf life estimation of fresh date fruits. *Sensors*. 2023;23(16):7081 <https://doi.org/10.3390/s23167081>

20. Capogrosso L, Cunico F, Cheng DS, Fummi F, Cristani M. A machine learning-oriented survey on Tiny Machine Learning. IEEE Access. 2024;12:3365349. <https://doi.org/10.1109/ACCESS.2024.3365349>
21. Banbury CR, Zhou W, Fedorov I, Matas R, Thakker U, Gope D, et al. Benchmarking TinyML systems: Challenges and direction. arXiv preprint arXiv:2003.04821. 2020. Available in: <https://arxiv.org/abs/2003.04821>
22. Hymel S, Banbury CR, Montemerlo R, Reddi VJ. Edge Impulse: An MLOps platform for Tiny Machine Learning. arXiv preprint arXiv:2212.03332. 2023. Available in: <https://arxiv.org/abs/2212.03332>
23. Lê MT, Bouzid Y, Miloud B, Hajjaji MA, Bensrhair A. Efficient neural networks for Tiny Machine Learning: A comprehensive review. arXiv preprint arXiv:2311.11883. 2023. Available in: <https://arxiv.org/abs/2311.11883>
24. Ray PP. A review on TinyML: State-of-the-art and prospects. J King Saud Univ Comput Inf Sci. 2022;34(4):1595-623. <https://doi.org/10.1016/j.jksuci.2021.11.019>
25. Sze V, Chen Y, Yang T. Efficient processing of deep neural networks: A tutorial and survey. Proc IEEE. 2017;105(12):2295-329. <https://doi.org/10.1109/JPROC.2017.2761740>
26. Henan Hanwei Electronics Co. Ltd. Technical Data MQ-135 Gas Sensor. Hanwei Electronics. 2013. Available in: https://www.electronicoscaldas.com/datasheet/MQ-135_Hanwei.pdf
27. Biglari A, Tang W. A review of embedded machine learning based on hardware, application, and sensing scheme. Sensors. 2023;23(4):2131. <https://doi.org/10.3390/s23042131>
28. MathWorks. nntraintool Neural network training tool - MATLAB. 2025 Feb 19 [cited in Mar 2025]. Available in: <https://www.mathworks.com/help/deeplearning/ref/nntraintool.html>
29. Bagheri M, Farshfroush N, Bagheri K, Shemirani AI. Applications of artificial intelligence technologies in water environments: From basic techniques to novel tiny machine learning systems. Process safety and environmental Protection. 2023;180(1):10-22. doi: <https://doi.org/10.1016/j.psep.2023.09.072>
30. Espressif Systems. ESP32 series datasheet. Espressif Systems. 2025 Feb 19 [cited in Mar 2025]. Available in: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf
31. Lourakis MIA. A Brief Description of the Levenberg-Marquardt Algorithm Implemented by levmar. Institute of Computer Science, Foundation for Research and Technology - Hellas (FORTH); Heraklion, Greece; 2005. Disponible en: <https://users.ics.forth.gr/lourakis/levmar/levmar.pdf>
32. Aljohani JL, Alaidaorous ES, Raja MAZ, Alhothuali MS, Shoaib M. Backpropagation of Levenberg-Marquardt artificial neural networks for wire coating analysis in the bath of Sisko fluid. Ain Shams Engineering Journal. 2021; 12(4): 4133-4143. doi: <https://doi.org/10.1016/j.asej.2021.03.007>
33. Ciuffoletti A. Low-Cost IoT: A Holistic Approach. Journal of Sensor and Actuator Networks. 2018; 7: 19. <https://doi.org/10.3390/jsan7020019>
34. Cubic Instruments. Portable Headspace Analyzer (MAP) Gasboard-3210Plus. 2025 Mar 5 [citado 2025 Mar 5]. Available in: <https://www.gasanalyzers.com/ProcessGasAnalyzer/info190>
35. Anton Paar. Alcohol and Extract Meter for the Beverage Industry. 2025 Mar 5 [cited in Mar 2025]. Available in: <https://www.anton-paar.com>