



UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE CUENCA

CARRERA DE INGENIERÍA MECÁNICA AUTOMOTRIZ

**DESARROLLO DE UN SISTEMA INTELIGENTE DE DIAGNÓSTICO DE FALLAS
MECÁNICAS EN MOTORES DE ENCENDIDO PROVOCADO A TRAVÉS DEL
SENSOR MAP PARA DISPOSITIVOS MÓVILES EN PLATAFORMA ANDROID**

Trabajo de titulación previo a la obtención del
título de Ingeniero Mecánico Automotriz

AUTOR: CHRISTIAN ROBERTO ESCANDÓN PESÁNTEZ

TUTOR: ING. RAFAEL WILMER CONTRERAS URGILÉS

Cuenca - Ecuador

2024

**CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE
TITULACIÓN**

Yo, Christian Roberto Escandón Pesántez con documento de identificación N° 0106652100, manifiesto que:

Soy el autor y responsable del presente trabajo; y, autorizo a que sin fines de lucro la Universidad Politécnica Salesiana pueda usar, difundir, reproducir o publicar de manera total o parcial el presente trabajo de titulación.

Cuenca, 28 de febrero del 2024

Atentamente.



.....
Christian Roberto Escandón Pesántez

0106652100

**CERTIFICADO DE CESIÓN DE DERECHOS DE AUTOR DEL TRABAJO DE
TITULACIÓN A LA UNIVERSIDAD POLITÉCNICA SALESIANA**

Yo, Christian Roberto Escandón Pesántez con documento de identificación N° 0106652100, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Proyecto técnico: “Desarrollo de un sistema inteligente de diagnóstico de fallas mecánicas en motores de encendido provocado a través del sensor MAP para dispositivos móviles en plataforma Android”, el cual ha sido desarrollado para optar por el título de: Ingeniero Mecánico Automotriz, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En concordancia con lo manifestado, suscribimos este documento en el momento que hacemos la entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.

Cuenca, 28 de febrero del 2024

Atentamente.



.....
Christian Roberto Escandón Pesántez

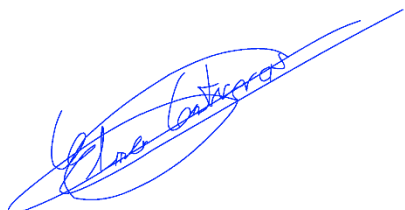
0106652100

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Rafael Wilmer Contreras Urgilés con documento de identificación N° 0104463328, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DESARROLLO DE UN SISTEMA INTELIGENTE DE DIAGNÓSTICO DE FALLAS MECÁNICAS EN MOTORES DE ENCENDIDO PROVOCADO A TRAVÉS DEL SENSOR MAP PARA DISPOSITIVOS MÓVILES EN PLATAFORMA ANDROID, realizado por Christian Roberto Escandón Pesántez, con documento de identificación N° 0106652100, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 28 de febrero del 2024

Atentamente.



.....
Ing. Rafael Wilmer Contreras Urgilés

0104463328

AGRADECIMIENTO

Principalmente, agradezco a mi madre Blanca por guiarme siempre hacia el camino del bien, apoyándome y brindándome todo su amor incondicional.

Gracias a toda mi familia. En especial a mi hermana Andrea por todo su apoyo y cuidados a lo largo de mi vida, a mi hermano Bryam por su ayuda en todos mis momentos difíciles, a mi abuelita Teresa por sus consejos.

Mis agradecimientos a mis sobrinos Justin y Dylan por toda la felicidad y amor que me han entregado desde el momento en el que nacieron.

Mi agradecimiento a los mejores amigos de mi vida, Óscar y Maya, por compartir su vida conmigo y enriquecer mi vida con bonitas experiencias.

A la Universidad Politécnica Salesiana por brindarme la oportunidad de ser un profesional.

A mi director de tesis el ingeniero Wilmer Contreras, por su apoyo, experiencia y motivación, para lograr desarrollar este proyecto, agradezco su amistad, sinceridad y confianza.

Christian Roberto Escandón Pesantez.

DEDICATORIA

El proyecto de titulación se lo dedico con cariño y agradecimiento a mi madre, Ing. Blanca Pesantez, quién con un arduo sacrificio, ha hecho posible la culminación de esta etapa en mi vida, y quien con su amor, perseverancia y apoyo incondicional hace de mí una mejor persona.

Christian Roberto Escandón Pesantez.

INDICE GENERAL

AGRADECIMIENTO	V
DEDICATORIA	VI
RESUMEN	1
ABSTRACT	2
INTRODUCCIÓN	3
PROBLEMA	4
ANTECEDENTES	4
IMPORTANCIA Y ALCANCES	4
DELIMITACION	5
OBJETIVOS	5
Objetivo general.....	5
Objetivos Específicos	6
1. CAPITULO I: Revisión bibliográfica y Fundamentos teóricos.	7
1.1. Estado del arte	7
1.2. Fundamentos teóricos	14
1.2.1. Motor	14
1.2.2. Motor de combustión interna	14
1.2.3. Ciclo operativo ideal	15
1.2.4. Ciclo operativo real	17
1.2.5. Motor de inyección electrónica	19
1.2.6. Diagnóstico de fallas	25
1.2.7. Metodología estadística	25
1.3. Machine learning	26
1.3.1. Random Forest	27
1.3.2. Redes neuronales	28
1.4. RASPBERRY PI	29
1.4.1. Características principales	30
1.5. Convertidor analógico digital ADS1115	30
1.6. Python	32
1.6.1. Características principales de Python	32
1.6.2. Ventajas de Python	33
1.7. LabVIEW	34
1.8. Matlab	34
1.9. Comunicación MQTT	34

1.10.	Plataforma Android	35
1.10.1.	Características Fundamentales	35
2.	CAPITULO II: Marco metodológico.....	36
2.1.	Unidad experimental de datos adquiridos	36
2.2.	Fallas provocadas	39
2.2.1.	Fallas provocadas de un cilindro al desconectar el inyector	39
2.2.2.	Fallas provocadas ajustando la apertura del electrodo de la bujía.....	40
2.2.3.	Fallas provocadas en las bobinas de encendido	41
2.2.4.	Fallas provocadas en sistema de distribución	42
2.2.5.	Fallas provocadas al filtro de aire.	43
2.2.6.	Factores no controlados del experimento	44
2.3.	Toma de muestras, creación de base de datos	44
2.3.1.	Conexión y configuración de la raspberry pi	44
2.3.2.	Caracterización del sensor MAP	44
2.3.3.	Esquema de conexión sensor – raspberry.....	45
2.3.4.	Protocolo para la adquisición de datos para la creación de base de datos.	47
2.3.5.	Análisis de varianza	50
2.4.	Diseño del sistema inteligente de diagnóstico en plataforma Android.....	53
2.4.1.	Creación y entrenamiento del algoritmo de predicción.	53
2.4.2.	Diseño del protocolo de comunicación del sistema de predicción.	55
2.4.3.	Diseño del aplicativo Android.....	58
2.5.	Funcionamiento del Sistema inteligente de diagnóstico	61
3.	CAPITULO III: Análisis de resultados.....	62
3.1.	Resultado de clasificación mediante algoritmo de bosque aleatorio	62
3.2.	Resultados del sistema de diagnóstico inteligente de fallas mecánicas.....	62
3.3.	Validación del sistema de diagnostico	63
3.3.1.	Validación de los datos muestrales	63
3.3.2.	Validación de resultados en la clasificación de las fallas mecánicas	64
3.4.	Conclusiones	66
3.5.	Recomendaciones	67
4.	Bibliografía.....	68
5.	ANEXOS	71
5.1.	ANEXO 1	71
5.2.	ANEXO 2	77
5.3.	ANEXO 3	79

5.4. ANEXO 4	82
---------------------------	----

INDICE DE FIGURAS

Figura 1	Ubicación de la Universidad Politécnica Salesiana.	5
Figura 2	Diagrama del ciclo Otto ideal.	16
Figura 3	Ciclo operativo de los motores MEC.	17
Figura 4	Ciclo real de los motores MEP.	18
Figura 5	Diagrama de distribución motores MEP.	19
Figura 6	Tanque de combustible.	20
Figura 7	Bomba de inyección.	21
Figura 8	Riel de combustible.	22
Figura 9	Inyectores.	22
Figura 10	Sensor MAP.	23
Figura 11	Pines del sensor MAP.	24
Figura 12	Centralita electrónica.	24
Figura 13	Random forest.	28
Figura 14	Esquema de una red neuronal.	29
Figura 15	Convertidor A/D ADS1115.	31
Figura 16	Interfaz de software Python.	33
Figura 17	Logo Android.	35
Figura 18	Panel de control del banco didáctico.	37
Figura 19	Interruptores de los inyectores.	40
Figura 20	Calibración de bujías.	41
Figura 21	Interruptor de bobina #2 #3.	41
Figura 22	Interruptor de bobina #1 #4.	42
Figura 23	Calibración de distribución.	42
Figura 24	Filtro de aire tapado al 50 %.	43
Figura 25	Filtro tapado al 75%.	43
Figura 26	Curva característica del sensor.	45
Figura 27	Esquema de conexión sensor - raspberry.	45
Figura 28	Broadcom de los pines GPIO de la raspberry.	46
Figura 29	Flujograma para la obtención de señales.	47
Figura 30	Matriz de datos estadísticos.	49
Figura 31	Graficadora LabVIEW.	49
Figura 32	Resultante de error, red de prueba.	50
Figura 33	Grafica de residuos para la variable estadística MODA.	52
Figura 34	Grafica de intervalos de MODA vs RESPUESTA.	52
Figura 35	Comparación de parejas de Tukey de la variable estadística MODA.	52
Figura 36	Flujograma de creación y entrenamiento del algoritmo.	53
Figura 37	Fragmento 1 de código, del algoritmo de entrenamiento.	54
Figura 38	Fragmento 2 de código, del algoritmo de entrenamiento.	55
Figura 39	Interfaz Hive MQ.	56
Figura 40	Diagrama de funcionamiento de la comunicación del sistema.	56
Figura 41	Fragmento 3 de código, del algoritmo de entrenamiento.	57
Figura 42	Fragmento 4 de código, del algoritmo de entrenamiento.	57
Figura 43	Fragmento 5, cálculo del error.	58

Figura 44	Interfaz de usuario en el aplicativo Android.	58
Figura 45	Explicación de funcionamiento del sistema, pasos 1 y 2.	59
Figura 46	Explicación de funcionamiento del sistema, paso 3.	60
Figura 47	Explicación de funcionamiento del sistema, paso 4.	60
Figura 48	Explicación de funcionamiento del sistema, paso 5.	61
Figura 49	Diagrama de funcionamiento del algoritmo.	62
Figura 50	Grafica de residuos de la Mediana.	64
Figura 51	Grafica de residuos de la Moda.	64
Figura 52	Graficas de residuos del sistema de diagnóstico.	65
Figura 53	Valor de p y r cuadrada.	65
Figura 54	Grafica de residuos para atributo, Máximo.	71
Figura 55	Grafica de comparación de parejas de Tukey, para atributo máximo.	71
Figura 56	Grafica de residuos para atributo, Media.	72
Figura 57	Grafica de comparación de parejas de Tukey, para atributo media.	72
Figura 58	Grafica de residuos para atributo, Mediana.	73
Figura 59	Grafica de comparación de parejas de Tukey, para atributo mediana.	73
Figura 60	Grafica de residuos para atributo, mínimo.	74
Figura 61	Grafica de comparación de parejas de Tukey, para atributo mínimo.	74
Figura 62	Grafica de residuos para atributo, moda.	75
Figura 63	Grafica de comparación de parejas de Tukey, para atributo moda.	75
Figura 64	Grafica de residuos para atributo, RMS.	76
Figura 65	Grafica de comparación de parejas de Tukey, para atributo RMS.	76
Figura 66	Fragmento 1, código del proceso de diagnóstico.	77
Figura 67	Fragmento 2, código del proceso de diagnóstico.	78
Figura 68	Fragmento 3, código del proceso de diagnóstico.	79
Figura 69	Fragmento 1, código de construcción y entrenamiento del algoritmo.	80
Figura 70	Fragmento 2, código de construcción y entrenamiento del algoritmo.	80
Figura 71	Fragmento 3, código de construcción y entrenamiento del algoritmo.	80
Figura 72	Fragmento 4, código de construcción y entrenamiento del algoritmo.	81
Figura 73	Fragmento 5, código de construcción y entrenamiento del algoritmo.	81
Figura 74	Fragmento 1, código de construcción del aplicativo.	82
Figura 75	Fragmento 2, código de construcción del aplicativo.	82
Figura 76	Fragmento 3, código de construcción del aplicativo.	83
Figura 77	Fragmento 4, código de construcción del aplicativo.	83
Figura 78	Fragmento 5, código de construcción del aplicativo.	83
Figura 79	Fragmento 6, código de construcción del aplicativo.	84
Figura 80	Fragmento 7, código de construcción del aplicativo.	84

INDICE DE TABLAS

Tabla 1 Características técnicas de la raspberry.	29
Tabla 2 Características técnicas del chip ADS1115.	30
Tabla 3 Identificación de pines ADS 1115	31
Tabla 4 Especificaciones técnicas del banco didáctico.....	36
Tabla 5 Resultados de las revisiones previas.	38
Tabla 6 Codificación de fallas a simular.....	39
Tabla 7 Caracterización sensor MAP.....	45
Tabla 8 Identificación del cableado del sensor MAP.	47
Tabla 9 Resultados del motor en falla 1300.....	62

RESUMEN

Este proyecto de grado tiene como objetivo desarrollar un sistema de diagnóstico inteligente en plataforma Android, mediante la aplicación de un algoritmo clasificador de Random forest, que permite detectar fallas mecánicas en los motores de encendido provocado, en base a la recolección y procesamiento de la señal de voltaje del sensor MAP.

Inicialmente, para el desarrollo del sistema de diagnóstico se procede a la obtención de la señal del sensor MAP, con una raspberry Pi con sistema operativo Raspbian y el software Python. Esta señal se adquiere con una alta confiabilidad del motor Hyundai Sonata de cuatro cilindros 2000 cm³ a gasolina, que se encuentra en la cámara semianecoica de la Universidad Politécnica Salesiana. En el motor se crean diferentes tipos de fallas mecánicas simuladas en elementos y sistemas como: sistema de distribución, bujías, inyectores, bobinas, filtros. La obtención de datos sigue un protocolo en el que se recopilan muestras cuando el motor experimenta una falla y cuando está en condiciones óptimas de funcionamiento; después de generar un fallo, se espera un tiempo de 5 minutos para que el motor se estabilice, consecuentemente se registran las señales por un tiempo de 3 minutos. Del procesamiento de estas señales se obtienen diferentes atributos estadísticos que permiten caracterizar la señal del sensor MAP y crear una base de datos para el entrenamiento del algoritmo; luego se desarrolla un aplicativo móvil capaz de controlar operaciones y visualizar resultados; a través de un protocolo de comunicación HTTP y el uso de la herramienta Hive MQ, se interconectaron todos los elementos del sistema de diagnóstico inteligente.

Mediante la aplicación del sistema, se adquirieron nuevos datos de un motor con similares características del que está en estudio, se realiza la evaluación del sistema de diagnóstico, para lo cual se genera una falla mecánica específica controlada. Posteriormente, desde el aplicativo se visualizan los resultados del diagnóstico y se verifica que el resultado del algoritmo corresponda la falla simulada; por último, se validan los resultados mediante los valores estadísticos P y R cuadrada, del análisis ANOVA, por el cual se determina que el sistema tiene un margen de confiabilidad del 98.5%, para la detección de fallas mecánicas que se presenten en el funcionamiento de un motor de encendido provocado.

Palabras clave: Random forest, MAP, sistema de diagnóstico inteligente, sensores automotrices, Hive MQ, comunicación HTTP.

ABSTRACT

This degree project aims to develop an intelligent diagnosis system on the Android platform, through the application of a Random Forest classifier algorithm, which allows detecting mechanical failures in spark ignition engines, based on the collection and processing of the signal. MAP sensor voltage.

Initially, for the development of the diagnostic system, the MAP sensor signal is obtained with a Raspberry Pi with Raspbian operating system and Python software. This signal is acquired with a high reliability of the Hyundai Sonata four-cylinder 2000 cm³ gasoline engine, which is located in the semi-anechoic chamber of the Salesian Polytechnic University. In the engine, different types of simulated mechanical failures are created in elements and systems such as: distribution system, spark plugs, injectors, coils, filters. Data collection follows a protocol in which samples are collected when the engine experiences a failure and when it is in optimal operating condition; After generating a fault, a time of 5 minutes is waited for the motor to stabilize, consequently the signals are recorded for a time of 3 minutes. From the processing of these signals, different statistical attributes are obtained that allow characterizing the MAP sensor signal and creating a database for training the algorithm; Then a mobile application is developed capable of controlling operations and visualizing results; Through an HTTP communication protocol and the use of the Hive MQ tool, all elements of the intelligent diagnostic system were interconnected.

Through the application of the system, new data were acquired from an engine with similar characteristics to the one under study, the evaluation of the diagnostic system is carried out, for which a specific controlled mechanical failure is generated. Subsequently, the diagnostic results are displayed from the application and it is verified that the result of the algorithm corresponds to the simulated failure; Finally, the results are validated using the statistical values P and R squared, from the ANOVA analysis, by which it is determined that the system has a reliability margin of 98.5%, for the detection of mechanical failures that occur in the operation of an engine ignition trigger.

Keywords: Random forest, MAP, intelligent diagnostic system, automotive sensors, HTTP communication.

INTRODUCCIÓN

El diagnóstico de anomalías en el motor de combustión interna se realiza con la combinación de técnicas de diagnóstico rápido mediante los sentidos de tacto, oído y visión, a su vez se aplican técnicas intrusivas y nocivas para vehículo. Este proceso implica las capacidades competentes del técnico de mantenimiento para el correcto análisis, sin embargo, estas acciones pueden llegar a causar daños en los componentes auxiliares del motor, lo que provoca un costo de reparación superior para el usuario y retrasos en la entrega de vehículos para el operario.

En la actualidad debido a los avances tecnológicos, el uso de sistemas inteligentes ha aumentado exponencialmente dentro del campo automotriz. Según (Leon Japa y Maldonado Ortega) se ha demostrado que esta herramienta posee una alta precisión en el diagnóstico y detección de fallas en los motores de combustión interna. Este proceso se utiliza con el objetivo de disminuir costo de reparación para el usuario y disminuir el tiempo de detección e identificación de fallas en el motor de combustión interna.

El desarrollo de este trabajo de titulación está enfocado en generar una herramienta de diagnóstico inteligente de falla mecánica en el MCI, mediante la aplicación de un algoritmo de machine learning, por medio de la recolección y análisis de la señal del sensor MAP. Para lo cual es necesario desarrollar una aplicación móvil para visualizar resultados y controlar esta herramienta.

PROBLEMA

El motor de combustión interna puede presentar fallas dentro de su desempeño que no impida su funcionamiento, sin embargo, compromete su eficacia. A causa de esto, los usuarios eluden el proceso de diagnóstico, lo que conlleva a que el motor trabaje con el fallo. Debido a esto el desgaste del motor es mayor, lo cual aumenta los costos de reparación y el tiempo para el proceso.

En la actualidad existen fallas mecánicas en los motores de combustión interna (MEP) que no pueden ser identificadas por la ECU, lo que provoca la ausencia de reconocimiento del scanner automotriz. En estos casos, se debe realizar un diagnóstico intrusivo del motor para localizar el fallo, en el cual se aplican técnicas tales como; la detección de fugas, medir compresión, depresión en el motor, presión de combustible, presión de aceite, entre otras. A través de estos procesos se ralentizan los tiempos de diagnóstico debido a que depende de la subjetividad del operario.

ANTECEDENTES

Actualmente, los motores de combustión interna tienen sistemas de elevada complejidad por su evolución en el tiempo. En este transcurso, se han desarrollado piezas de alta precisión las cuales se desgastan con el funcionamiento de este, lo que provoca fallas mecánicas en el motor de combustión interna. Por ende, es preciso diagnosticar estas fallas de manera adecuada con la mínima aplicación de técnicas intrusivas.

IMPORTANCIA Y ALCANCES

Este proyecto tiene como finalidad aportar a la carrera de Ingeniería Mecánica Automotriz de la Universidad Politécnica Salesiana, en específico al Grupo de Investigaciones de Ingeniería de Transporte GIIT, y materias a fines al diagnóstico de motores de combustión interna. Mediante este proyecto se pretende desarrollar un sistema de diagnóstico de fallas mecánicas en los MEC a través de análisis de señales del sensor MAP.

Esta técnica es mínimamente invasiva y de alta eficacia y rapidez, que emplea los patrones de la señal eléctrica del sensor MAP mediante el uso de sistemas inteligentes, que no se rigen bajo subjetividades de los operarios. El sistema de diagnóstico permite identificar la falla en los motores de combustión interna, específicamente en el motor Hyundai Sonata, que se encuentra en la cámara semianecoica de la Universidad Politécnica Salesiana.

DELIMITACION

Este proyecto de titulación se llevará a cabo en la Provincia del Azuay, Ciudad de Cuenca, sur del Ecuador, a una altitud de 2550 m.s.n.m, una extensión de 70.59 km² y una población aproximada de 580000 habitantes. En el taller de la Carrera de Ingeniería Mecánica Automotriz de la Universidad Politécnica Salesiana, se realiza una parte práctica que consta de la recopilación de información de la maqueta didáctica del motor “Hyundai Sonta 2.0 DOCH”. En la figura 1 se detalla la georreferencia del proyecto.

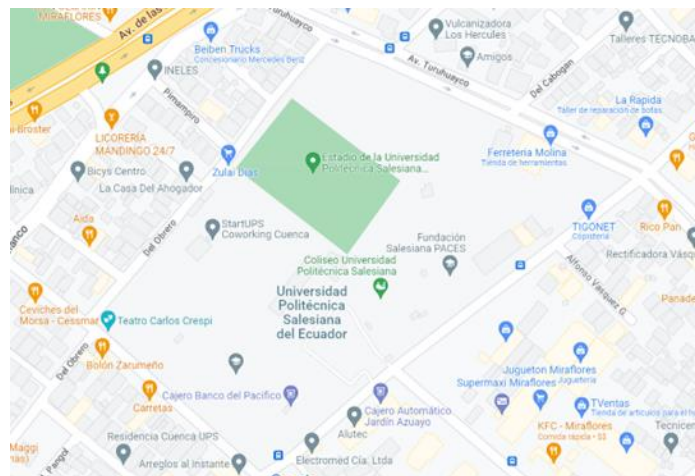


Figura 1 Ubicación de la Universidad Politécnica Salesiana.

Fuente: Google Maps

GRUPO OBJETIVO

El proyecto va dirigido como material adicional para los estudiantes y profesores de la carrera de Ingeniería Automotriz de la Universidad Politécnica Salesiana sede Cuenca.

OBJETIVOS

Objetivo general

Desarrollar un sistema inteligente de diagnóstico de fallas mecánicas en motores de encendido provocado a través del sensor MAP para dispositivos móviles en plataforma Android.

Objetivos Específicos

Revisar el estado del arte mediante una búsqueda de información relacionada en bases de datos institucionales o gubernamentales, para obtener información sobre las variables representativas y las condiciones de funcionamiento de los motores MEP y el sensor MAP.

Establecer y aplicar las variables y condiciones de funcionamiento optimas mediante los resultados obtenidos en la revisión del estado del arte para el desarrollo del sistema de toma de datos y diagnóstico de fallas.

Diseñar e implementar un sistema de toma de datos y diagnóstico de fallas mecánicas basado en plataforma Android.

Analizar y concluir la precisión de los resultados obtenidos con el funcionamiento de la herramienta de diagnóstico en plataforma Android, a través de un análisis de varianza de los resultados.

1. CAPITULO I: Revisión bibliográfica y Fundamentos teóricos.

1.1. Estado del arte

El artículo "Determination of combustion parameters using engine crankshaft speed", publicado en el año 2013, aborda la importancia de los controles electrónicos de motor en el tiempo real para diagnosticar el proceso de combustión, crucial para cumplir con regulaciones ambientales y eficiencia de combustible más estrictas. Se destaca la complejidad y el costo intrusivo de medir directamente la presión en el cilindro, lo que lleva a la propuesta de utilizar una red neuronal de perceptrón multicapa para modelar la relación entre la velocidad del cigüeñal y parámetros derivados de la presión en el cilindro. El modelo de red neuronal permite estimar de manera no intrusiva la presión del cilindro y evaluar en tiempo real la calidad de la combustión. Se detalla la estructura del modelo y su procedimiento de entrenamiento, sugiriendo además un controlador de combustión basado en la información extraída de la velocidad del cigüeñal. La aplicación del modelo en un motor de encendido por chispa de un solo cilindro, sometido a diversas velocidades y cargas, confirma la eficacia en la estimación de parámetros de presión de la combustión mediante un procesamiento adecuado de la señal de velocidad del cigüeñal. En las conclusiones destaca la precisión del modelo de red neuronal al predecir parámetros clave de la curva de presión en el cilindro, mostrando correspondencia con los valores medidos en diversas condiciones. Se sugiere que este modelo puede ser una herramienta crucial para sistemas avanzados de control de motores, incluso proponiendo un sistema de control en bucle cerrado que utiliza la ubicación angular del pico de presión para modificar en tiempo real el ángulo de avance de la chispa, mejorando así la eficiencia y el rendimiento del motor (Taglialatela, Vaglieco y Mancaruso).

El artículo "Desarrollo de un algoritmo para el diagnóstico del estado porcentual del rendimiento de los cilindros, mediante el análisis de la señal del sensor de posición del cigüeñal de un motor Nissan FD 6 cilindros" publicado en el año 2014, presenta un algoritmo de diagnóstico innovador para evaluar el rendimiento de combustión y compresión de los cilindros de un motor diésel. Utilizando la velocidad instantánea del motor medida por el sensor de posición del cigüeñal, el método se fundamenta en la reconstrucción completa de la curva instantánea del par efectivo para un ciclo del motor, empleando un modelo dinámico del cigüeñal. El enfoque se destaca por utilizar el sensor CKP (Crankshaft Position Sensor) como herramienta principal para el diagnóstico de motores de combustión interna (MCIA). Aunque atractivo por su independencia de sensores adicionales costosos, como los utilizados en análisis por vibraciones, sonido o termografía, su complejidad de señal y requisitos de software

especializado han limitado su implementación hasta ahora a equipos de diagnóstico de fabricantes de vehículos. El sensor CKP, capaz de medir las fluctuaciones en la velocidad angular instantánea del cigüeñal, ha permitido el desarrollo de métodos "no intrusivos" para la detección de fallas de combustión. Los resultados de las pruebas de combustión, evaluadas mediante el algoritmo propuesto, revelan que los cilindros con fallas tienden a sobrecargar a los cilindros con mejor rendimiento. Además, se destaca que un rendimiento superior en uno o varios cilindros no garantiza su buen estado, ya que podrían estar enfrentando problemas como exceso de combustible o compensando las fallas en otros cilindros. El criterio propuesto para determinar la salud general del motor sugiere que el rendimiento de todos los cilindros debe superar el 80%, y la diferencia máxima entre cilindros no debe exceder el 10%. Este criterio también puede aplicarse a pruebas de compresión, ofreciendo un enfoque integral para el diagnóstico del estado de los cilindros y del motor en general (Hurtado Piña, Morocho Cabrera y Pingüil Lliguicota).

El artículo "Diseño e Implementación de un Sistema para Determinar Fallas Mecánicas en Motores de Encendido Provocado Mediante Redes Neuronales Artificiales" publicado en el año 2017, se aborda el desarrollo de un sistema basado en redes neuronales artificiales (RNA) para identificar fallas mecánicas en motores de encendido provocado (MEP). Utilizando señales de los sensores KS y CMP en un motor Hyundai SONATA 4 cilindros 2.0 DOHC a gasolina, se generan condiciones de fallas mecánicas intencionadas en bobinas, inyectores, bujías, sistema de distribución, presión de combustible, filtro de aire y catalizador. Las señales se procesan con filtros virtuales, y los datos se introducen en Matlab para crear la RNA. El sistema se evalúa con nuevas muestras del motor, generando fallas mecánicas específicas y utilizando LabVIEW para analizar los resultados. Se aplica un análisis de diseño experimental (ANOVA) para validar la confiabilidad del sistema, concluyendo que la RNA tiene un alto porcentaje de confiabilidad del 95% en la detección de fallas mecánicas en el funcionamiento del MEP. Entre las conclusiones destacadas, se resalta la aplicabilidad del sistema para diferentes condiciones de funcionamiento del motor Hyundai SONATA 2.0, permitiendo identificar fallas mecánicas de manera poco invasiva sin desarmar el MEP. Se enfatiza la correcta interpretación de resultados mediante ANOVA, evidenciando la confiabilidad del sistema en la visualización de fallas mecánicas. Se mencionan desafíos asociados a fallas en el sistema de distribución y se destaca la eficacia del algoritmo en LabVIEW para detectar fallas mecánicas en MEP en tan solo 2 minutos. La elección de una RNA específica, con 73 neuronas y 146 capas ocultas, se respalda por su eficiencia en generar respuestas en un tiempo máximo

de 2 horas durante el entrenamiento. La compatibilidad entre LabVIEW y Matlab se reconoce como una ventaja, permitiendo una aplicación eficaz de la RNA. Se concluye que la implementación del sistema de RNA, basado en la señal del sensor KS, logra reconocer diversos tipos de fallas mecánicas en el MEP con errores mínimos, respaldando la eficacia del sistema implementado (Garzon Iñiguez y Urdiales Baculima).

El artículo "Sistema de Diagnóstico Automotriz Mediante el Análisis de Emisiones Contaminantes con la Aplicación de Redes Neuronales para la Detección de Fallas" publicado en el año 2017, aborda el desarrollo de un sistema inteligente de diagnóstico automotriz basado en redes neuronales. El objetivo principal es mejorar el mantenimiento automotriz utilizando datos en tiempo real de las emisiones contaminantes como base de conocimiento. El sistema tiene como entrada las emisiones contaminantes y como salidas diversas fallas del motor, como la apertura del inyector, apertura del electrodo de bujías y presión de la bomba de combustible. El análisis se centra en tres motores diferentes: Sonata 2.0, Samsung SM7 V6 y Chevrolet LUV. Se destaca la importancia de realizar el diagnóstico a 1600 RPM para exponer el sistema a la mayor variabilidad, considerando la sensibilidad de las emisiones contaminantes a ruidos. La normalidad en la toma de datos se respalda mediante análisis de gráficas de residuos y histogramas, que muestran un comportamiento de correlación positiva en las emisiones de los tres motores. El estudio identifica variables más significativas, como la presión de la bomba, la apertura del inyector y la falla en la apertura de electrodos de bujías. Se sugiere la generación de módulos independientes para cada tipo de motor con sus respectivas bases de conocimiento. Las interacciones de las variables de falla resultan significativas, y se descarta la posibilidad de una base de conocimiento compartida para los tres motores. El sistema de diagnóstico muestra eficiencia en la detección de fallas, con la inclusión de lambda y AFR para mejorar la precisión del diagnóstico. La red neuronal Newff se identifica como la más eficiente, con un error de detección de fallas medio de 3.38×10^{-5} . Sin embargo, se señala una dificultad en la detección de ciertas condiciones de falla, como un inyector al 75%, lo que puede deberse a calibraciones diferentes o a la generación de fallas previas al diagnóstico. A pesar de estos desafíos, el sistema de diagnóstico automotriz demuestra eficacia, siempre que se minimicen las variables de ruido (Torres Vargas).

El artículo "Identificación de Patrones de Fallas Mecánicas Mediante Redes Neuronales Artificiales para el Diagnóstico de Motores de Encendido Provocado" publicado en el año 2018, propone un sistema de diagnóstico basado en redes neuronales artificiales (RNA) para detectar fallas mecánicas en motores de encendido provocado. Utilizando señales de sensores

MAP y CMP obtenidas del motor Hyundai Sonata 4 cilindros 2.0 cc, se generan intencionalmente diversas anomalías mecánicas en componentes como bobinas, inyector, bujías, presión de combustible y sistema de distribución. El proceso de obtención de muestras se realiza en condiciones ideales de temperatura y ralentí, registrando las señales después de inducir una falla y estabilizar el motor. Se aplican diferentes métodos estadísticos para analizar atributos de las señales MAP, se seleccionan los más influyentes para el entrenamiento de la RNA. La evaluación del sistema se lleva a cabo mediante la introducción de nuevas muestras con fallas mecánicas controladas, es importante verificar la correspondencia entre los resultados de la RNA y las fallas intencionadas. La validación se realiza mediante análisis de varianza (ANOVA), la cual demuestra que la RNA tiene una confiabilidad del 96.5% en la detección de fallas mecánicas en motores de encendido provocado. En las conclusiones, se destaca el cumplimiento del objetivo general, que confirma que el sistema de diagnóstico basado en RNA puede detectar patrones de fallas mecánicas en este tipo de motores. Se comparan diferentes métodos de entrenamiento de la RNA, se selecciona la configuración con el menor porcentaje de error. La interfaz Labview-Matlab se elogia por su facilidad y precisión en la aplicación de redes neuronales en el diagnóstico automotriz. La validación estadística muestra la equivalencia entre la respuesta de clasificación real y los resultados de la RNA, con un error de clasificación cercano al 0.00%. El artículo subraya la eficacia de las redes neuronales como herramienta valiosa para el diagnóstico automotriz, proporciona un enfoque rápido y mínimamente invasivo. El diseño experimental factorial 2² se emplea para garantizar la coherencia en la experimentación, y se utilizan diversos métodos estadísticos para identificar los atributos más relevantes en la diferenciación de patrones de falla (Leon Japa y Maldonado Ortega).

El artículo "Determinación de la Presión Máxima de Compresión de un Motor de Encendido Provocado basado en una Red Neuronal Artificial Recurrente" publicado en el año 2018, presenta una metodología para estimar la presión máxima de compresión (MEP) en motores de combustión interna de encendido provocado (MEP). La investigación se centra en caracterizar las curvas de consumo de amperaje del motor de arranque como base para la determinación de la MEP. Se utiliza un protocolo de adquisición de datos y análisis estadístico, el cual selecciona valores como energía, promedio, desviación estándar, varianza, kurtosis, asimetría, máximo, mínimo y factor de cresta para la caracterización del experimento. Los valores estadísticos generan bases de datos que se aplican en la creación y entrenamiento de una red neuronal artificial recurrente (RNAR), que logra un error absoluto inferior al 2%. La

metodología se prueba inicialmente en un motor en un banco didáctico y luego se aplica a motores de vehículos. Los resultados destacan que la aplicación de redes neuronales recurrentes para determinar la compresión en motores MEP es una alternativa viable y mínimamente invasiva, con un error menor al 1%. La técnica se presenta como una opción eficiente para integrarse en sistemas de diagnóstico, que aprovechan la rapidez computacional de las redes neuronales. La investigación resalta la idoneidad de la estructura de red neuronal tipo Elman para este propósito, debido a la naturaleza dinámica de los patrones obtenidos del análisis del consumo de corriente del motor de arranque. En conclusión, la metodología propuesta no solo ofrece una estimación precisa de la MEP, crucial en evaluaciones de compraventa de vehículos, sino que también destaca la utilidad de las redes neuronales en diagnósticos eficientes y rápidos en el campo automotriz (Contreras y Arichavala).

En el artículo "Diagnosis of Sensor Faults in a Combustion Engine Control System with the Artificial Neural Network" publicado en el año 2019, se investiga un motor de combustión interna de encendido por chispa con inyección directa de gasolina, sometiéndolo a condiciones simuladas de fallos en diversos sensores. A pesar de que el sistema de diagnóstico a bordo no detectó daños, la configuración del sistema de control del motor se ajustó dinámicamente mediante un algoritmo adaptativo en respuesta a fallos simulados en sensores de temperatura del aire, temperatura del refrigerante, presión del combustible, presión del aire, fuga en el colector de admisión y perturbaciones en el flujo de aire. Se recopilaron datos de señales de todos los sensores disponibles y del sistema de diagnóstico a bordo. Con esta gran cantidad de datos, se creó una función estadística para clasificar fallos en sensores que utilizan una red neuronal artificial. El conjunto de datos de entrenamiento se preparó en consecuencia, y durante el aprendizaje, la red neuronal alcanzó una tasa de aciertos de más del 99%. Las conclusiones destacan la robustez del sistema de control del motor, que se reconfigura dinámicamente para adaptarse a fallos individuales en sensores sin interrupciones significativas. A pesar de que la indicación de los sensores no superó los límites establecidos, el sistema de control adaptativo ajustó sus parámetros para optimizar el rendimiento y la eficiencia del motor en cada caso. Se señala que, en algunos casos, la adaptación puede no ser beneficiosa, especialmente cuando el motor opera con mezclas pobres, las que dificultan la detección de desviaciones anómalas, como la reducción de potencia. Para abordar esta problemática, se desarrolló un sistema de diagnóstico de fallos en sensores basado en una red neuronal con una capa oculta. Los resultados satisfactorios, con una tasa de aciertos superior al 99%, respaldan la eficacia del método de aprendizaje automático para identificar fallos en sensores. Se destaca la

automatización del procedimiento de diagnóstico gracias al uso de métodos de inteligencia artificial, los que permiten la clasificación de fallos sin necesidad de reglas específicas y adaptándose a algoritmos adaptativos propios de cada sistema de control. Además, se sugiere que otros métodos de aprendizaje automático, como árboles de decisión o máquinas de soporte vectorial, podrían aplicarse con éxito en futuras investigaciones (Komorska, Wolczynski y Borczuch).

El artículo "Desarrollo de un Prototipo Computacional de Diagnóstico de Averías Mediante Recopilación de Datos del Sensor CKP de un Motor Daewoo GM F16D3A" publicado en el año 2021, describe un sistema de diagnóstico de averías en el motor GM F16D3A del Chevrolet AVEO mediante el uso de redes neuronales artificiales (RNA) basadas en datos recopilados del sensor CKP (Crankshaft Position Sensor). La herramienta de diagnóstico se centra en analizar la velocidad angular del cigüeñal durante la fase de combustión del ciclo Otto, convertida en una señal de voltaje por el sensor CKP. El protocolo de adquisición de datos sigue un enfoque estrictamente definido, lo que permite la correcta obtención y análisis de parámetros estadísticos clave de la señal del sensor CKP. Estos parámetros, como área, máximo, mínimo, varianza, kurtosis, raíz media cuadrada (RMS), factor de cresta, asimetría, energía y potencia, son significativos para el diagnóstico. Los datos se procesan con un algoritmo en LabVIEW para crear y entrenar una RNA tipo Newff (feed-forward backpropagation) con función de transferencia "traincgb". La validación del sistema se realiza mediante análisis estadístico ANOVA, que tiene una fiabilidad mayor al 95%. El sistema determina patrones de comportamiento del motor en condiciones de avería supervisada con éxito. Se selecciona la configuración de RNA con menor error tras diferentes entrenamientos. El estudio de ANOVA valida los resultados de la red frente a la condición del motor, lo que demuestra similitud entre los valores reales y los obtenidos por la RNA. La fase de comprobación revela un error esperado en la detección de averías, especialmente en los sensores O2 y TPS, aunque se considera dentro del rango de error aceptable. En conclusión, los métodos no intrusivos de diagnóstico, como las redes neuronales, son destacados por su utilidad en el diagnóstico automotriz, que optimiza recursos y complementa las técnicas de mantenimiento convencionales (Luna Avila y Tapia Palacios).

El artículo "Identificación del Fallo de los Inyectores de un MEC mediante el Análisis de las Señales de los Sensores CKP y CMP utilizando RNA" publicado en el año 2022, presenta un sistema de diagnóstico para identificar fallas en los inyectores de un motor Hyundai Santa Fe 2.0 CRDi mediante el análisis de las señales de los sensores CKP (Crankshaft Position

Sensor) y CMP (Camshaft Position Sensor) utilizando Redes Neuronales Artificiales (RNA). El proceso comienza con una revisión bibliográfica que destaca la confiabilidad del sensor CKP para evaluar el estado del motor de manera no invasiva. Luego, se adquieren las señales de los sensores CKP y CMP que utilizan una tarjeta DAQ 6212, que asegura que el motor esté a ralentí y a una temperatura óptima de funcionamiento. Se induce una falla intencional en los inyectores, y las señales adquiridas se analizan y filtran con el software LabVIEW. Posteriormente, se genera una matriz de datos estadísticos basada en la variación de la frecuencia en el dominio del tiempo de cada cilindro. Esta matriz se utiliza en Matlab para crear y entrenar una red neuronal artificial (RNA) tipo Newff con la función de transferencia "trainscg", que logra un bajo error del 0.4176%. La validación del sistema se lleva a cabo mediante la obtención de nuevas muestras de los sensores del motor con fallas supervisadas en los inyectores. La interfaz de LabVIEW verifica que los resultados de la RNA coincidan con las fallas supervisadas. La validación del sistema se respalda con un análisis estadístico por el método de la varianza (ANOVA), la cual concluye que la RNA tiene una fiabilidad superior al 98% en la identificación del inyector con falla en los MEC (Motores de Encendido por Compresión). En conclusión, el sistema desarrollado, basado en RNA y utiliza las señales de los sensores CKP y CMP, demuestra ser eficaz en identificar inyectores con fallas en motores, por lo que cumple con éxito el objetivo general del proyecto. La aplicación de RNA en el diagnóstico automotriz, especialmente en los sensores del motor, destaca por su rapidez, objetividad y mínima invasión, que logra resultados en aproximadamente 5 minutos y reduciéndose a 10 segundos con un motor instrumentado (Calderon Orellana).

El artículo "Multiple Fault Detection and Isolation Using Artificial Neural Networks in Sensors of an Internal Combustion Engine" publicado en el año 2023, aborda la creciente complejidad en la detección de fallas en vehículos con motores de combustión interna debido al rápido avance tecnológico y al aumento en el uso de sensores y actuadores para mejorar el rendimiento del motor. Las herramientas especializadas para la detección de fallas a veces presentan errores en múltiples sensores cuando solo hay un problema en el sistema y no son responsables de aislar las fallas. Por ende, se requieren algoritmos superiores para la detección y aislamiento de fallas (FDI) en motores de combustión interna. La investigación se enfoca en diseñar un sistema de detección y aislamiento de fallas individuales y múltiples basado en redes neuronales artificiales (ANN) para sensores clave como el de posición del acelerador (TPS), flujo de aire masivo (MAF) y presión absoluta del colector (MAP). El sistema utiliza cinco ANNs, las que estiman los valores de dos sensores mediante la velocidad del motor y la señal

de relación aire-combustible (AFR), generando redundancia analítica. Cuando ocurre una falla en un sensor, el sistema de FDI sustituye la señal defectuosa con una estimación saludable proporcionada por una ANN. En caso de una segunda falla en otro sensor, el sistema reemplaza la señal defectuosa con la estimación de la ANN correspondiente, permitiendo que el motor continúe en funcionamiento incluso con múltiples fallas. Las conclusiones resaltan el diseño de un sistema de detección y aislamiento de fallas que, basándose en las estimaciones proporcionadas por las ANNs, detecta las fallas individuales y múltiples sin alterar el comportamiento del sistema. El esquema identifica y reemplaza sensores defectuosos con estimaciones aceptables, lo que permite que el motor continúe en funcionamiento a pesar de las fallas (García Morales, Cervantes Bobadilla y Saavedra Benítez).

1.2. Fundamentos teóricos

1.2.1. Motor

El motor se define como una máquina creada para convertir la energía potencial de una fuente, como puede ser un combustible, en energía mecánica para generar movimiento. Hay varios tipos de motores, como pueden ser los motores de combustión interna, motores eléctricos, motores de vapor, estos entre los más notables.

1.2.2. Motor de combustión interna

El motor de combustión interna tuvo un predecesor que fue la máquina de vapor. Como innovación revolucionaria, permitió el desarrollo del ferrocarril y sirvió como fundamento para la Revolución Industrial. En 1769, Cugnot en Francia implementó una caldera de vapor en un carro militar, con dos cilindros verticales impulsando una rueda a una velocidad máxima de 9,5 km/h y una autonomía limitada. En Estados Unidos, Evans patentó en 1789 un carruaje de vapor con tres cilindros, que construyó en 1803. Simultáneamente, el ingeniero británico Trevithick creó el primer carruaje de vapor en 1801, evolucionando en 1803, seguido por Gurney en 1829 y Carret en 1861.

El motor alternativo de combustión interna (MACI), basado en el mecanismo pistón-biela-manivela, surge a finales del siglo XIX como una alternativa a las máquinas de vapor para propulsar vehículos ligeros, las cuales reemplazaron carruajes y tracción animal. A pesar de varios intentos anteriores, la máquina de vapor demostró limitaciones en términos de rendimiento y tamaño adecuado (Álvarez Flórez y Callejón Agramunt).

El motor de combustión interna convierte la energía química del combustible en energía mecánica a través de un proceso de combustión dentro de una cámara interna. Estos motores

se emplean en diversos campos como automóviles, aviones, generadores de electricidad y herramientas. Los motores de combustión interna alternativos (MCIAs) son motores térmicos de desplazamiento positivo, que obtienen trabajo mediante el desplazamiento lineal del émbolo en un mecanismo biela-manivela. Estos se denominan así porque el estado térmico se genera en el fluido que evoluciona en el motor (Rovira de Antonio y Muñoz Domínguez, Máquinas y motores térmicos: introducción a los motores alternativos y a las turbomáquinas térmicas).

El trabajo de un motor de combustión interna se interpreta a través de ciclos operativos. Los ciclos operativos son dos, ciclo operativo ideal y ciclo operativo real. Ambos ayudan a la interpretación del proceso de trabajo de un motor de combustión interna.

1.2.3. Ciclo operativo ideal

El ciclo ideal que se observa en la figura 2, describe el funcionamiento de un motor de combustión interna de gasolina se conoce como ciclo Otto.

Este ciclo teórico consta de cuatro etapas fundamentales: admisión, compresión, combustión y escape. Aunque el ciclo Otto es asociado principalmente con motores de encendido por chispa, difiere significativamente del ciclo real, pero su utilidad radica en explicar el proceso bajo hipótesis simplificadas.

Como se ve en la figura 2, las etapas termodinámicas del ciclo Otto incluyen:

- Compresión adiabática (1-2): En esta etapa, el fluido de trabajo se comprime adiabáticamente, con el pistón realizando el trabajo de compresión (W_1).
- Aportación de calor a volumen constante (2-3): Se introduce instantáneamente el calor aportado (Q_1) a volumen constante.
- Expansión adiabática (3-4): La expansión tiene lugar adiabáticamente, y el fluido de trabajo realiza el trabajo de expansión (W_2).
- Extracción de calor a volumen constante (4-1): Se produce la extracción instantánea de calor (Q_2) a volumen constante. En los motores de 4 tiempos reales, la extracción de calor ocurre durante la fase de escape (4-1-0), y el fluido ingresa en la carrera de admisión (0-1).

En el ciclo Otto ideal, se representa gráficamente en el diagrama P/V de la figura 2, omitiendo los procesos 1-0 y 0-1, que se teóricamente anulan entre sí, lo que resulta en una pérdida o ganancia de calor nula (Álvarez Flórez y Callejón Agramunt).

El ciclo Otto es particularmente adecuado para máquinas reciprocantes de encendido por chispa. Nombrado en honor a Nikolaus A. Otto, quien construyó una máquina de cuatro tiempos en 1876 en Alemania, este ciclo es relevante para las máquinas de combustión interna de cuatro tiempos, donde el pistón ejecuta cuatro tiempos completos dentro del cilindro y el cigüeñal completa dos revoluciones por cada ciclo termodinámico (Moreno Navarrete).

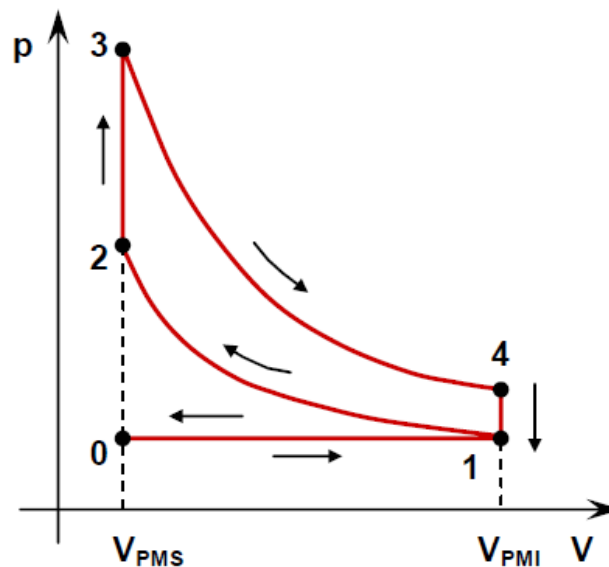


Figura 2 Diagrama del ciclo Otto ideal.

Fuente: (Vélez de Guevara)

A continuación, se explican las cuatro etapas fundamentales del ciclo de trabajo del motor de combustión interna.

1.2.3.1. Admisión

En esta fase, el pistón desciende del punto muerto superior al punto muerto inferior del cilindro, simultáneamente la válvula de admisión se abre para que el aire pueda entrar al cilindro debido a la depresión que produce el movimiento del pistón dentro del mismo. En la figura 3a se representa la etapa de admisión (Alonso Perez).

1.2.3.2. Compresión

Para esta etapa el pistón se encuentra en el punto muerto inferior. Simultáneamente la válvula de admisión se cierra y el pistón inicia la carrera del punto muerto inferior al punto muerto superior, en donde la mezcla aire-combustible que ingresó al cilindro, se comprime hasta llegar al volumen de compresión, directo de la cámara de combustión. En la figura 3b se representa la etapa de compresión (Alonso Perez).

1.2.3.3. Combustión

En este trayecto con la mezcla comprimida dentro de la cámara, la bujía genera una chispa eléctrica que enciende la mezcla. Esto provoca una explosión controlada dentro del cilindro y un momento de fuerza que impulsa al pistón del punto muerto superior al punto muerto inferior y además transmite la energía al cigüeñal para generar movimiento. En la figura 3c se representa la etapa de combustión (Alonso Perez).

1.2.3.4. Escape

Al finalizar la etapa de combustión, la válvula de escape se abre, la presión dentro del cilindro se iguala a la presión atmosférica y el movimiento del pistón al ascender del punto muerto inferior al punto muerto superior expulsa los gases resultantes de la combustión, que se encuentran dentro del cilindro a través de la válvula de escape. En la figura 3d se representa la etapa de escape (Alonso Perez).

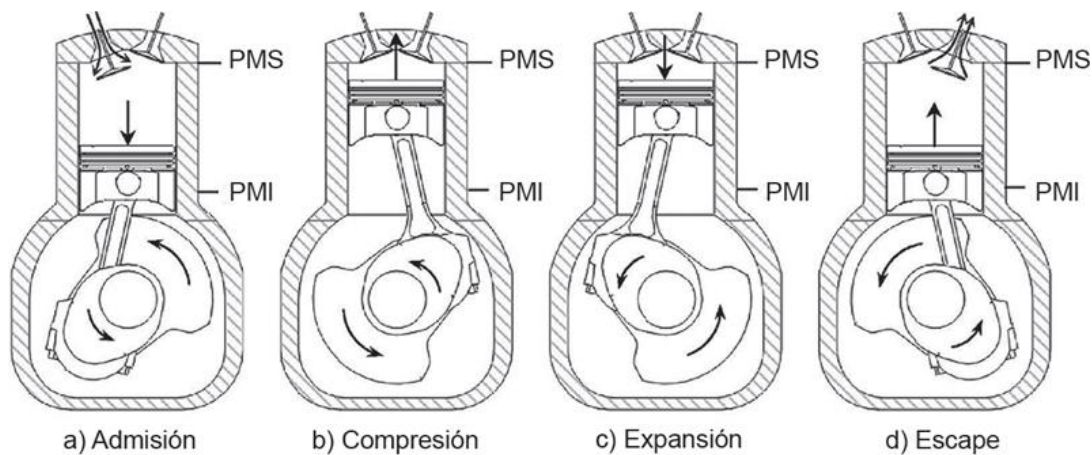


Figura 3 Ciclo operativo de los motores MEC.

Fuente: (Rovira de Antonio y Muñoz Domínguez, *Máquinas y motores térmicos: introducción a los motores alternativos y a las turbomáquinas térmicas*)

1.2.4. Ciclo operativo real

El ciclo de trabajo en los motores de combustión interna se aprecia en la figura 4, y se concibe tanto teórica como prácticamente, cuando se considera las imperfecciones inherentes a la operación práctica, como la pérdida de eficiencia debida a factores como el calor y la fricción. Estas deficiencias no solo impactan la eficiencia general del motor, sino que también contribuyen a la emisión de gases contaminantes, lo que resalta las limitaciones de la mezcla aire-combustible (Rovira de Antonio y Muñoz Domínguez, *Máquinas y motores térmicos: introducción a los motores alternativos y a las turbomáquinas térmicas*).

En contraste con el diagrama de presión y volumen teórico, que asume transformaciones perfectas, la realidad presenta diversos factores que afectan la cantidad de trabajo útil aportada por el motor. Aspectos críticos incluyen: la no llenura completa del cilindro a presión atmosférica durante la carrera de admisión, la no instantaneidad en la entrada de la mezcla de aire y gasolina al abrirse las válvulas, el retardo en la combustión después del salto de chispa en la bujía y la generación de presión por parte de los gases de escape al final de la fase de escape.

Para abordar estas limitaciones, se implementan modificaciones en el motor durante el ciclo real. Dada la imposibilidad de accionar las válvulas de forma instantánea, se realizan ajustes en los momentos de inicio y cierre durante los desfases para considerar la duración de los procesos de renovación de la carga. Estas adaptaciones se aprecian en el diagrama de la figura 5, y se manifiestan claramente en puntos específicos del ciclo real, como el Avance a la Apertura de Admisión (AAA) y el Avance a la Apertura de Escape (AAE), representando el proceso de admisión real. Estas alteraciones buscan mitigar los problemas originados por los factores mencionados, para contribuir a mejorar la eficiencia y reducir las emisiones contaminantes en la práctica de los motores de combustión interna (Moreno Navarrete).

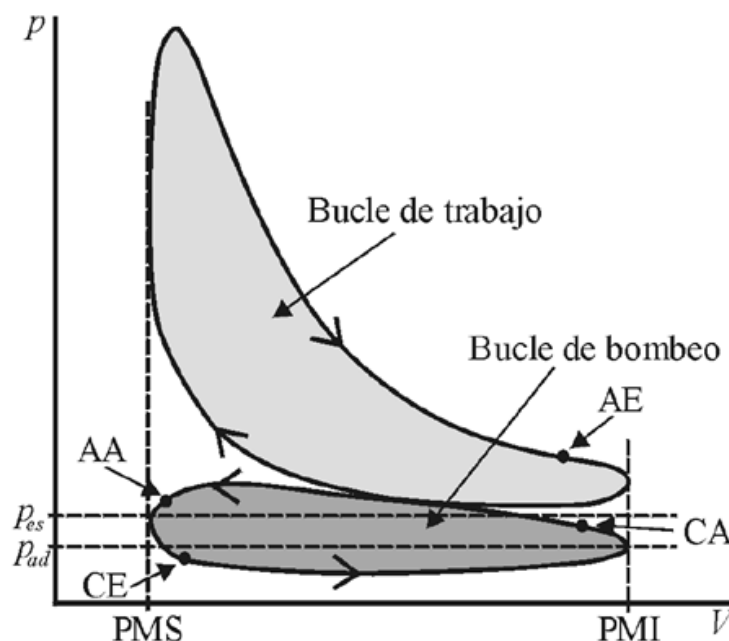


Figura 4 Ciclo real de los motores MEP.

Fuente: (Anonimo)

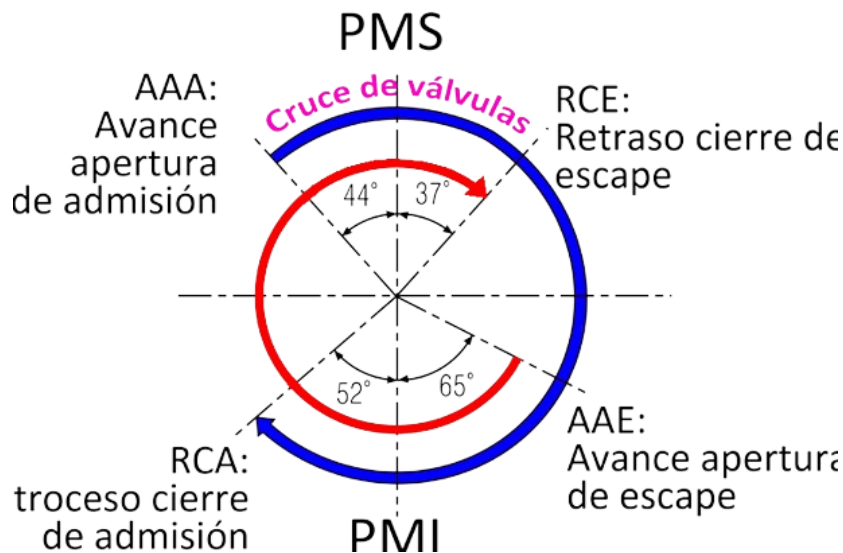


Figura 5 Diagrama de distribución motores MEP.

Fuente: (microscopio.pro)

El motor de combustión interna dentro de su proceso de trabajo tiene carencias y limitaciones. Para optimizar el trabajo del motor de combustión interna con la precisa aplicación de las modificaciones durante el ciclo real y minimizar el impacto de factores como fricción y calor, se desarrollan sistemas de inyección, que usando sensores cuantifica variables físicas y ofrece condiciones de funcionamiento aptas para el motor.

1.2.5. Motor de inyección electrónica

El motor de inyección electrónica utiliza un sistema de inyección de combustible, el cual es operado por una unidad electrónica de control (ECU) para manejar el flujo de combustible hacia las cámaras de combustión del motor.

Mediante la utilización de sensores se cuantifican las variables físicas necesarias y ofrecer las correctas condiciones de funcionamiento para el trabajo del motor, variables como la cantidad de oxígeno en los gases de escape, la cantidad de aire que ingresa, la presión en el colector de admisión. Estos datos se reciben y procesan en la ECU para ajustar la cantidad de combustible inyectado. El arranque de un motor empieza con el movimiento de los pistones del motor, estos suben y bajan simultáneamente, los sensores empiezan a indicar a la ECU la rotación del motor lo que, a su vez, permite la apertura de las válvulas de admisión y se suministra a la cámara la cantidad de combustible ideal para el volumen de aire admitido, lo cual genera una relación aire-combustible para realizar la combustión. Esta inyección de combustible se suministra de dos formas, directa o indirectamente, que posee como principal característica la presión de inyección. (Álvarez Flórez y Callejón Agramunt)

1.2.5.1. Elementos del motor de inyección electrónica

El funcionamiento de un motor de inyección electrónica requiere de elementos actuadores que, no son distintos en funcionamiento, pues su principal cambio es un accionamiento electrónico.

Estos elementos se enlistan a continuación:

- Tanque de Combustible
- Bomba de Combustible
- Riel de Combustible
- Inyectores de Combustible
- Sensores
- Centralita Electrónica (ECU)

1.2.5.2. Tanque de combustible

El tanque de combustible como se observa en la figura 6, es un contenedor diseñado para almacenar y suministrar combustible a un motor de combustión interna, se asegura un suministro constante para el funcionamiento adecuado del vehículo. Estos tanques de combustible tienen una capacidad de entre 15 y 30 galones para vehículos medianos y camionetas.



Figura 6 Tanque de combustible.

Fuente: (Spectra Premium)

1.2.5.3. Bomba de inyección

La bomba de inyección como se observa en la figura 7, es un componente clave del sistema de suministro de combustible, debido a que proporciona la presión necesaria para inyectar el combustible a la cámara de manera controlada y precisa en todos los cilindros del motor. La bomba de inyección en motores de inyección electrónica es accionada a través

señales electrónicas enviadas desde la ECU, lo que permite un control más preciso sobre la cantidad y el momento de la inyección de combustible, a esto se suma que la gestión electrónica ajusta dinámicamente la cantidad de combustible inyectado según las condiciones y necesidades del motor.

Funciona a una presión típica de 2 a 4 bar, suele tener un amperaje que oscila entre 5 y 20 amperios, con un flujo de aproximadamente de 80 a 120 litros por hora. Este dispositivo asegura un suministro constante y preciso de combustible al motor, crucial para mantener la eficiencia y rendimiento óptimos. Además, su diseño incluye tecnologías como motores eléctricos eficientes y materiales resistentes para garantizar durabilidad y fiabilidad en diversas condiciones de operación del vehículo.



Figura 7 Bomba de inyección.

Fuente: (Bosch)

1.2.5.4. Riel de combustible

El riel de combustible como se muestra en la figura 8, es un conducto que distribuye y almacena combustible antes de ser inyectado en los cilindros del motor, garantizando una entrega precisa para una combustión eficiente. Los rieles de combustible son fabricados de materiales resistentes a la corrosión y capaces de soportar la presión del combustible, los materiales más utilizados para su fabricación son el aluminio y el acero inoxidable.



Figura 8 Riel de combustible.

Fuente: (Bosch)

1.2.5.5. Inyectores

El inyector que se muestra en la figura 9, es el elemento encargado de introducir combustible en la cámara de combustión del motor de una manera precisa y controlada. Posee un funcionamiento similar al de una válvula que se abre y cierra a través de impulsos eléctricos comandados por la ECU. El inyector pulveriza una fina bruma de combustible que ingresa a la cámara de combustión debido a la succión que produce. Esta atomización del combustible es esencial para facilitar una mezcla homogénea de aire y combustible, lo que produce una mejora para la eficiencia de la combustión y reduce las emisiones. Cada inyector de combustible está equipado con una bobina electromagnética que se activa para abrir y cerrar el inyector, lo que permite el flujo controlado de combustible. El parámetro característico de la bobina electromagnética es el valor de resistencia común, que oscila entre 10 y 16 ohm.

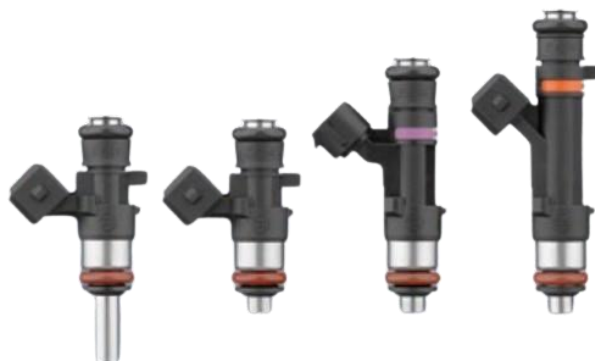


Figura 9 Inyectores.

Fuente: (Bosch)

1.2.5.6. Sensores

Un sensor se define como un elemento capaz de identificar una magnitud física para transformarla en una magnitud eléctrica, la cual es procesada por la unidad de control, para gestionar el correcto funcionamiento del motor.

Uno de los sensores empleados y más importantes dentro del sistema de inyección electrónica es el sensor MAP. Su relevancia radica en que la señal de voltaje que proporciona es una señal muy representativa del rendimiento del motor.

- **Sensor MAP**

El sensor MAP (Manifold Absolute Pressure) se visualiza en la figura 10, y mide la presión en el colector de admisión, emitiendo una señal eléctrica a la unidad de control del motor (ECU) para ajustar la inyección de combustible. En aplicaciones automotrices, se evolucionaron desde conceptos aeroespaciales costosos a diseños más económicos. Los sensores actuales no miden directamente la presión, sino el desplazamiento de un diafragma flexible por la presión del múltiple. Una opción económica es el sensor con calibrador de deformación difusa de diafragma de silicio, que utiliza una pastilla de silicio de 3 mm². La piezorresistividad en ciertos semiconductores provoca cambios proporcionales en la resistencia al aplicar presión en el múltiple (Álvarez Flórez y Callejón Agramunt).



Figura 10 Sensor MAP.

Fuente: (*Denso*)

. La salida del sensor MAP es típicamente una señal eléctrica analógica de voltaje variable. A medida que la presión en el colector experimenta cambios, la señal del sensor también varía. El esquema eléctrico básico del sensor MAP se ilustra en la figura 11.

- (A)Alimentación: se suministra con voltaje constante, con 5 voltios.
- (B)Tierra: Conexión a tierra para cerrar el circuito eléctrico, con valores menores a 60 mV.

- (D)Salida: Conexión de salida que proporciona una señal de voltaje variable en función de la presión en el colector. Esta señal se envía al módulo de control del motor (ECU). Los sensores MAP poseen un rango de presión que oscila entre 0 a 20 psi.
- (C)Señal compartida: señal de temperatura (IAT) agregada en el encapsulado del sensor.



Figura 11 Pines del sensor MAP.

Fuente: (Denso)

1.2.5.7. Centralita Electrónica (ECU)

La Centralita Electrónica (ECU) que se visualiza en la figura 12, es el cerebro del sistema de inyección de un motor de combustión interna, que procesa datos de sensores para controlar la inyección de combustible, el encendido y otros parámetros, que optimizan así el rendimiento del motor. Ajusta dinámicamente la mezcla aire-combustible y la sincronización del encendido para adaptarse a las condiciones de conducción y garantizar eficiencia y potencia óptimas.



Figura 12 Centralita electrónica.

Fuente: (Amazon)

Después del análisis los componentes del sistema de inyección electrónica, se examinan los conceptos del diagnóstico de fallas para una mejor comprensión, interpretación e identificación de las fallas mecánicas que se presentan en los motores MEP.

1.2.6. Diagnóstico de fallas

El diagnóstico automotriz de fallas se refiere a la búsqueda, identificación y resolución de problemas en vehículo, para reconocer su causa en una maquina o sistema. El diagnostico implica el uso y aplicación de herramientas de valoración para adquirir datos para un eficaz registro de información.

Los errores son encontrados mediante métodos de detección y técnicas de diagnóstico, se aplican según las características dinámicas del sistema y el número de señales disponibles por variable. Existen muchos métodos de detección de fallas, tales como: Chequeo de límites y detección de cambios, chequeos de tendencias, estimación de media, varianza, pruebas estadísticas, análisis para señales periódicas, filtros pasa banda, análisis de Fourier, transformada de Fourier, análisis de vibración de máquinas, entre otras.

Los resultados obtenidos en la revisión del estado del arte se basan en el estudio estadístico de la señal, por lo que se revisan los métodos estadísticos aplicados.

1.2.7. Metodología estadística

La metodología estadística ayuda en la compilación, organización y procesamiento de datos para conseguir conclusiones acertadas y resultados verídicos. La metodología estadística se divide en dos tipos: estadística descriptiva e inferencial.

1.2.7.1. Estadística inferencial

La estadística inferencial determina los aspectos relevantes en un grupo de datos, en base a una muestra la cual aplica procedimientos y métodos estadísticos, como por ejemplo el análisis ANOVA el cual indica el grado de fiabilidad de los resultados.

1.2.7.2. Análisis de varianza (ANOVA)

El análisis de varianza ANOVA, es un grupo de técnicas aplicadas para analizar la transición de una variable medida en situaciones definidas por factores discretos. Esto evalúa la importancia de uno o más factores al comparar las medias de las variables de respuesta en los diferentes niveles de factores. Las técnicas estadísticas son aplicables matemáticamente a factores lineales, pero para fenómenos más complejos se utilizan técnicas de inteligencia artificial como RNA.

1.2.7.3. Estadística descriptiva

La Estadística descriptiva se aplica para hallar los valores más representativos de cada señal al visualizar los datos de forma más comprensible lo que provoca una mejor interpretación. Por lo cual, mediante la revisión del estado del arte, se obtuvo los atributos estadísticos más útiles en el desarrollo de investigaciones para el diagnóstico de fallas en motores mediante la señal del sensor MAP, parámetros que se enlistan a continuación.

1. Mínimo
2. Máximo
3. Media
4. Mediana
5. Moda
6. Rango
7. Asimetría
8. Suma
9. Curtosis
10. Desviación estándar
11. Media cuadrática

Cuando la estadística descriptiva llega a su límite se opta por la aplicación de herramientas avanzadas de procesamiento de señales como el Machine learning.

1.3. Machine learning

Una rama de la inteligencia artificial (IA), es el Machine Learning (ML). El ML se centra en el desarrollo de algoritmos y modelos que posibilitan a las computadoras, para aprender patrones y realizar tareas sin ser programadas directamente. Los modelos de aprendizaje automático mejoran su rendimiento con el tiempo, adaptarse a nuevas situaciones para mejorar la precisión, y usar una varios métodos y enfoques de aplicación de Machine Learning según sea necesario.

Existen dos enfoques de desarrollo en el campo del Machine Learning, los cuales se dividen en aprendizaje supervisado y no supervisado. En el aprendizaje supervisado, se proporcionan ejemplos de capacitación, mientras que en el aprendizaje no supervisado se descubren patrones en el material sin etiquetar (Alaminos Fernández).

Además, existen métodos de ensamble que implican la combinación de múltiples modelos con el fin de mejorar la precisión y la robustez del sistema, al fusionar las predicciones de

varios modelos más simples, se reduce la susceptibilidad al sobreajuste y se mejora la capacidad de generalización. Los dos enfoques principales de ensamble son Bagging, que utiliza múltiples instancias del mismo algoritmo entrenadas en subconjuntos de datos diferentes, y Boosting, que asigna pesos a los modelos según su rendimiento, ajustándolos en interacciones sucesivas (Alaminos Fernández).

Debido a que se asocia la presente investigación con un proceso de clasificación de fallas, se determina que el algoritmo Random Forest es una opción viable para este estudio.

1.3.1. Random Forest

El algoritmo Random Forest es un método de ensamble desarrollado dentro del enfoque de Bagging. Este algoritmo crea un conjunto de árboles de decisión durante el entrenamiento. Al utilizar técnicas de Bagging se puede reducir la varianza y mejorar la precisión del modelo (Alaminos Fernández).

Random Forest, durante el entrenamiento construye un bosque de árboles de decisión como se ejemplifica en la figura 13. Cada árbol se entrena con un subconjunto aleatorio de datos y características, lo que reduce el riesgo de sobreajuste. Durante la predicción, los árboles emiten sus decisiones y la salida final se determina por votación o promedio. Esto mejora la precisión y la generalización del modelo (Alaminos Fernández).

Random Forest es un algoritmo versátil, resistente a valores atípicos y eficaz tanto en clasificación como en regresión. Además, proporciona medidas de la importancia de las características, lo que facilita la interpretación del modelo (Gonzalez).

Este algoritmo se encuentra dentro del enfoque de aprendizaje supervisado. El modelo se entrena al utilizar un conjunto de datos que contiene valores de entrada junto con las respuestas correctas asociadas, y luego utiliza estos árboles para hacer predicciones sobre nuevos datos no etiquetados (Data Science).

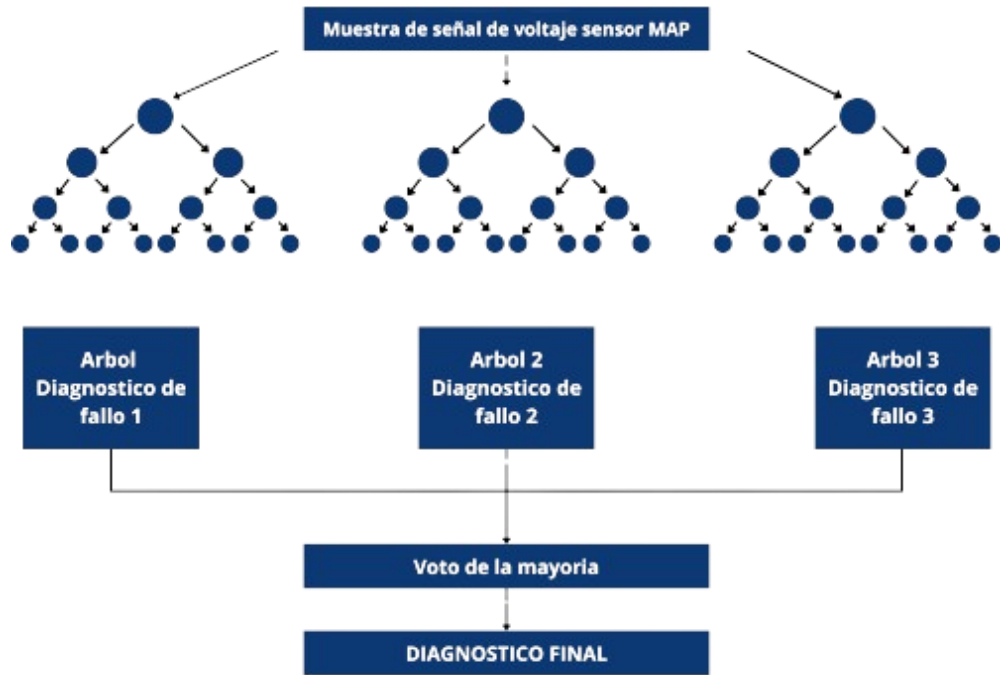


Figura 13 Random forest.

Fuente: Autor

1.3.2. Redes neuronales

Las redes neuronales como se ejemplifica en la figura 14, son sistemas de aprendizaje complejos basados en la estructura y funcionamiento del cerebro humano. Están formadas por nodos llamados neuronas artificiales, organizadas en capas, que reciben entradas, realizan cálculos y generan salidas. La información fluye a través de conexiones ponderadas, donde el peso indica la importancia. Durante el entrenamiento, los pesos se ajustan para mejorar la precisión en la predicción o clasificación de datos. Estas redes son versátiles, aplicándose en reconocimiento de imágenes, procesamiento de lenguaje natural y más. La inteligencia artificial, fundamental en aprendizaje automático, modela relaciones complejas y extrae patrones. Las definiciones varían, describiéndolas como computación inspirada en modelos biológicos o como sistemas matemáticos. Las redes neuronales, interconectadas masivamente, procesan información dinámicamente en respuesta a entradas externas, reflejando la organización jerárquica de la inteligencia artificial (Matich, 2001).

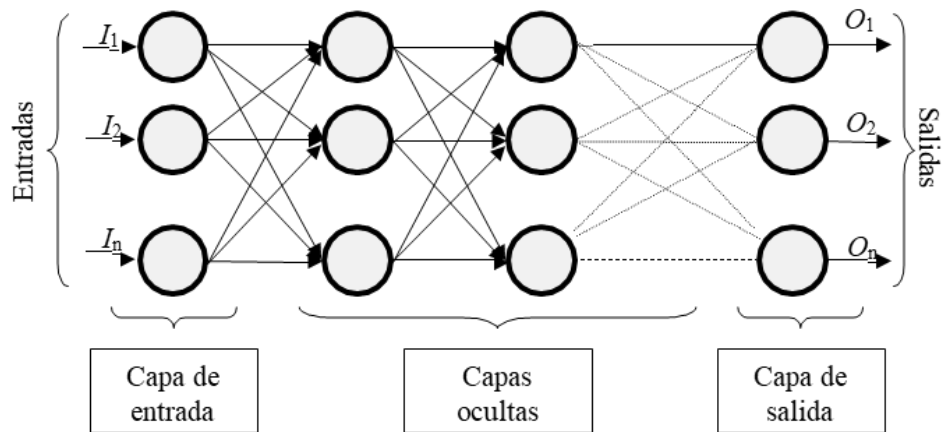


Figura 14 Esquema de una red neuronal.

Fuente: (Jimenez, Paz - Arias y Larco)

Se aplican redes neuronales para la validación de la base de datos, ya que esta herramienta determina si existe una correlación entre los atributos de la base de datos y la condición de funcionamiento del motor.

1.4. RASPBERRY PI

La Raspberry Pi es una computadora de placa única (SBC, por sus siglas en inglés) creada por la Raspberry Pi Foundation. Se define como placas diseñadas para ser asequibles, versátiles y de bajo consumo de energético. Una Raspberry Pi integra todos los componentes básicos de un sistema informático en una pequeña placa de circuito de características como la versatilidad y bajo costo, las características técnicas de la raspberry se ven en la tabla 1.

Tabla 1 Características técnicas de la raspberry.

Fuente: Autor

Definición	Características
Procesador	Broadcom BCM2711
	Core Cortex-A72 a 1.5 GHz
Memoria RAM	2GB
	4GB
	8GB LPDDR4
Almacenamiento	Según tarjeta SD usada
Conectividad	Ethernet Gigabit
	Wi-Fi 802.11ac
	Bluetooth 5.0
Puertos USB	2 puertos USB 3.0
	2 puertos USB 2.0

Video y sonido	2 puertos micro HDMI que admiten hasta 4K a 60 fps.
	Conector de audio de 3.5 mm y soporte para audio digital a través de HDMI.
GPIO	Encabezado de 40 pines que permite la conexión de periféricos y la expansión del hardware.
Alimentación	A través de un conector USB-C.
Sistema operativo	Compatible con una variedad de sistemas operativos, incluyendo Raspbian (basado en Debian) y otros sistemas operativos Linux.

1.4.1. Características principales

- Versatilidad: Se utiliza para una amplia gama de aplicaciones, desde proyectos educativos, de reparación, hasta servidores domésticos y estaciones de medios.
- Bajo Costo: La Raspberry Pi es asequible, lo que la hace accesible para estudiantes, aficionados y profesionales con presupuesto limitado.
- Comunidad Activa: Una comunidad en línea activa que comparte proyectos, soluciones y brinda soporte.
- Expansión: La presencia de pines GPIO permite la conexión de una variedad de sensores y periféricos, lo que facilita la expansión del hardware.
- Bajo Consumo de Energía: Diseñada para ser eficiente en términos de consumo de energía, lo que la hace ideal para proyectos alimentados por baterías.

Debido a que la señal de salida del sensor MAP, es una señal de voltaje analógica, para complementar la raspberry pi, se agregó un convertidor analógico digital ADS1115, el cual no compromete la calidad de recolección de datos y asegura la conexión de elementos.

1.5. Convertidor analógico digital ADS1115

El chip ADS1115 como se ve en la figura 15, es un convertidor analógico a digital de 16 bits. Tiene la capacidad de convertir señales analógicas en señales digitales con alta precisión y resolución. Cuenta con un protocolo de comunicación I2C, ideal en aplicaciones que requieren lecturas precisas de sensores analógicos, las características técnicas del ADS1115 se ven en la tabla 2.

Tabla 2 Características técnicas del chip ADS1115.

Fuente: Autor

Definición	Características
------------	-----------------

Resolución	16 bits
Canales de entrada	4 canales de entrada analógica
	2 canales en modo diferencial
Interfaz	I2C
Rango de voltaje de operación	5 V
	2 V
Tasa de conversión programable	Hasta 860 muestras por segundo
Amplificador de ganancia programable	2/3 x
	1x
	2x
	4x
	8x
	16x
Alarma de voltaje programable	Permite configurar límites superiores e inferiores para las alertas en los pines de ALERT/RDY.
Consumo de corriente	Bajo consumo de corriente en operación y modo de reposo.
Temperatura de operación	-40°C a +125°C

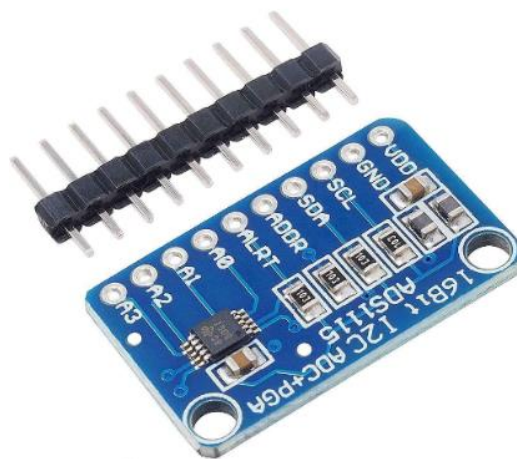


Figura 15 Convertidor A/D ADS1115.

Fuente: (Amazon)

Tabla 3 Identificación de pines ADS 1115

Fuente: Autor

Pin	Descripción
VDD	Suministro de energía 3v
GND	Tierra
I2C SDA Y SCL	SDA pin de datos
	SCL pin de reloj
ALERT/RDY	Configurado para alertas del chip
A0 a A3	Entradas analógicas, modo de entrada única y diferencial
ADDR	Configuración de ganancia

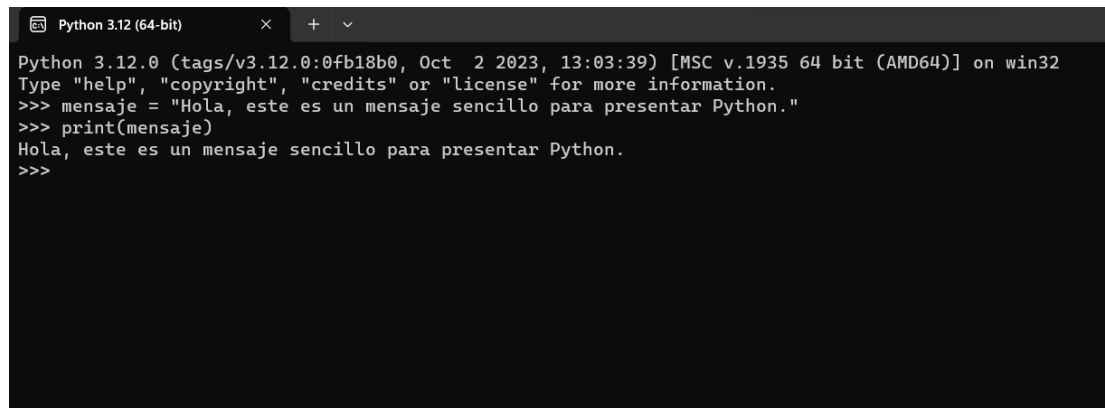
1.6. Python

Python es un lenguaje de programación de propósito general, interpretado y de alto nivel. Fue creado por Guido Van Rossum y la primera versión se lanzó en 1991. Python es conocido por su sintaxis clara y legible, lo que facilita la escritura y el mantenimiento del código. Es un lenguaje común que utiliza en una variedad de aplicaciones, desde desarrollo web hasta inteligencia artificial, en la figura 16, se puede ver un ejemplo de la interfaz de programación de Python.

1.6.1. Características principales de Python

Python posee una sintaxis simple y legible, lo que facilita la escritura y la comprensión del código del programa, este lenguaje de programación cuenta con algunas características como:

- **Multiparadigma:** Permite una programación funcional, imperativa y orientada a objetos, lo que ayuda a los desarrolladores elegir el método que mejor se adapte a sus necesidades.
- **Dinámico y fuertemente tipado:** Python es dinámico, lo que representa que las variables puedan cambiar de tipo durante la ejecución del programa. Sin embargo, también está fuertemente tipado, lo que produce que las conversiones de tipos necesiten ser precisas.
- **Amplia biblioteca estándar:** Python incluye una amplia gama de módulos y bibliotecas estándar que facilitan el desarrollo de una variedad de aplicaciones sin tener que empezar desde cero.



```
Python 3.12 (64-bit)
Python 3.12.0 (tags/v3.12.0:0fb18b0, Oct 2 2023, 13:03:39) [MSC v.1935 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> mensaje = "Hola, este es un mensaje sencillo para presentar Python."
>>> print(mensaje)
Hola, este es un mensaje sencillo para presentar Python.
>>>
```

Figura 16 Interfaz de software Python.

Fuente: Autor

1.6.2. Ventajas de Python

- Fácil de aprender y usar

La sintaxis simple y legible hace que Python sea adecuado para principiantes y su versatilidad lo convierte en una buena opción para desarrolladores experimentados.

- Comunidad activa

Python tiene una gran comunidad de desarrolladores que proporciona bibliotecas, marcos y recursos para facilitar el acceso a soluciones y ayuda.

- Versatilidad

Uso en una variedad de aplicaciones, desde desarrollo web y scripting hasta inteligencia artificial y análisis de datos.

- Desarrollo rápido

Las funciones interpretativas y de alto nivel de Python permiten un desarrollo rápido y eficiente.

- Soporte multiplataforma

Python es compatible con múltiples sistemas operativos, lo que facilita la portabilidad de las aplicaciones desarrolladas en el lenguaje.

En resumen, Python es un lenguaje de programación de propósito general que es fácil acceso y uso, el cual posee una gran comunidad y amplias aplicaciones en el desarrollo de software.

1.7.LabVIEW

LabVIEW, es un entorno de software para adquisición de datos, control de instrumentos y automatización de sistemas. Este se caracteriza por su interfaz gráfica y programación basada en bloques que permite el desarrollo de aplicaciones de monitoreo y control en tiempo real, así como el análisis y visualización de datos a través de gráficos personalizados. LabVIEW usa G como lenguaje de programación con fines visuales, donde los usuarios conectan bloques para definir el flujo del programa. Algunos de sus beneficios son la capacidad de ser flexible, adaptable e integrarse con varios sistemas de hardware. LabVIEW es una plataforma potente y fácil de usar que se utiliza ampliamente en sistemas de prueba, automatización industrial, investigación científica y desarrollo de productos para desarrollar soluciones de control y adquisición de datos.

1.8.Matlab

MATLAB, cuyo nombre completo es "MATrix LABoratory", es un entorno de software de alto rendimiento que su principal función es el cálculo numérico, simulaciones y visualización de datos. MATLAB es un lenguaje de programación y un entorno de desarrollo interactivo centrado en la manipulación de matrices, el análisis numérico y la resolución de problemas relacionados con la ingeniería y la ciencia. Se utiliza ampliamente en ingeniería, física, matemáticas aplicadas y otras disciplinas científicas.

Se aplican redes neuronales con MatLab, para la validación de la base de datos, ya que esta herramienta determina si existe una correlación entre los atributos de la base de datos.

1.9.Comunicación MQTT

La comunicación MQTT, que por sus siglas en inglés significa, Message Queuing Telemetry Transport, se define como un protocolo de comunicación ligero y eficiente, diseñado para la comunicación entre dispositivos en redes con ancho de banda limitado o en condiciones de conectividad inestable.

Basado en el modelo de publicación-suscripción, permite que los dispositivos publiquen mensajes en "temas" y otros se suscriban para recibirlos. Con niveles de calidad de servicio (QoS) que aseguran la entrega confiable, MQTT facilita la conexión entre dispositivos distribuidos.

Para este trabajo investigativo, se estudia la posible aplicación del protocolo MQTT, en la comunicación entre los elementos del sistema de diagnóstico como son la raspberry pi y el aplicativo Android.

1.10. Plataforma Android

La plataforma Android se creó por Android Inc, una empresa fundada por Andy Rubin, Rich Miner, Nick Sears y Chris White en octubre de 2003, el logo de este sistema operativo se visualiza en la figura 17, esta herramienta es de mucha importancia en el mundo de los sistemas operativos móviles. Android ha evolucionado desde su creación en 2003, brinda a los usuarios una experiencia diversa y productiva, lo cual provoca que Android sea una opción líder en el mercado.

1.10.1. Características Fundamentales

- Código abierto

Android es un sistema operativo de código abierto y su código fuente está disponible públicamente. Esto fomenta la colaboración y permite a los usuarios personalizar el sistema para adaptarlo a sus necesidades.

- Interfaz de usuario intuitiva: La interfaz de usuario de Android se caracteriza por su diseño intuitivo y fácil navegación. Los íconos, widgets y menús están organizados lógicamente para brindar una experiencia fluida y accesible para usuarios de esta plataforma a todos los niveles.
- Dispositivos: Android es compatible con una amplia gama de dispositivos, desde smartphones, tablets, televisores y relojes inteligentes. Esta diversidad permite a los usuarios elegir entre una variedad de opciones según sus preferencias.
- Personalización: Android tiene un alto nivel de personalización. Los usuarios deciden sobre los fondos de pantalla, instalar widgets, usar diferentes lanzadores de aplicaciones y personalizar la configuración según sus necesidades personales.



Figura 17 Logo Android.

Fuente: *(icon-icons)*

2. CAPITULO II: Marco metodológico.

Para el desarrollo de la presente sección se realiza los siguientes puntos:

- Unidad Experimental
- Fallas provocadas a la unidad experimental
- Recolección de datos
- Procesamiento de datos

2.1.Unidad experimental de datos adquiridos

El proyecto y su desarrollo se fundamenta en la metodología utilizada por (Caldas), por lo cual para la recolección de datos es necesario conocer características de la unidad experimental además de comprobar su estado de funcionamiento. El banco de información didáctico utilizado para el desarrollo de la investigación es “Gasoline IAG. A/T control system educational training equipment”, el cual cuenta con un motor a gasolina “Hyundai sonata 2.0 DOHC”. Sus características se especifican en la tabla 3.

Tabla 4 Especificaciones técnicas del banco didáctico.

Fuente: Autor

Característica	Valor
Cilindros	4
Cilindrada	2000
Accionamiento	DOHC
Relación de compresión	10.5:1
Potencia	175@6000 RPM
Par	168@4000
Combustible	Gasolina
Transmisión	Automática
Nombre del equipo	Banco didáctico del motor Hyundai sonata 2.0 DOHC
Modelo	G 110701
Dimensiones	250 X 80 X 180 cm
Peso	205 kg
Marca	DAE SUNG

El banco didáctico está equipado con un panel de control que permite simular fallas de sensores o actuadores mediante interruptores, en la figura 18 se observa el panel de control del banco didáctico.

El requerimiento previo al desarrollo de este proceso de investigación es realizar un proceso de mantenimiento correctivo y preventivo a los elementos y sistemas del motor Hyundai sonata 2.0 DOCH para la correcta obtención de señales.



Figura 18 Panel de control del banco didáctico.

Fuente: Autor

Para verificar el funcionamiento adecuado del motor de combustión interna, se genera una lista de parámetros, condiciones y comprobaciones de funcionamiento, con el fin de garantizar un proceso de experimentación adecuado a través de un óptimo funcionamiento. El proceso de mantenimiento previo garantiza el correcto desempeño de este, mediante el correcto funcionamiento del MEP se afirma que las señales emitidas por los sensores serán confiables y servirán para completar una base de datos y crear el algoritmo de clasificación. El mantenimiento general del motor incluyó la revisión de componentes del motor enlistados a continuación.

1. Estado de bujías
2. Estado de cables de bujía
3. Estado de filtro de aire
4. Estado de filtro de gasolina
5. Temperatura de funcionamiento del motor
6. Análisis con escáner OBD2
7. Estado de bomba de gasolina

8. Estado de inyectores
9. Nivel del aceite motor
10. Nivel del refrigerante del motor
11. Voltaje de la batería
12. Fugas de aceite o refrigerante

Una vez finalizado el proceso de mantenimiento general del motor, se comprueba su estado de funcionamiento el cual se resume en la tabla 4 de resultados de pruebas, por lo que se concluye que el motor se encuentra en óptimas condiciones.

Tabla 5 Resultados de las revisiones previas.

Fuente: Autor

REVISIÓN DE ESTADO MECANICO DEL MOTOR SONATA 2.0			
Parámetros	Valor	Estado	Observaciones
Estado de bujía		OK	
Estado de cables de bujía		OK	
Estado de bobinas de encendido		OK	
Estado de filtro de aire		OK	
Estado de filtro de gasolina		OK	
Temperatura de funcionamiento del motor	92°C	OK	Entre 90°C y 98°C
Estado de bomba de gasolina	43.3 psi	OK	
Análisis con escáner OBD 2		OK	No presenta DTC
Estado de inyectores		OK	Tiempo 2.16 ms
Nivel de aceite del motor		OK	
Nivel de refrigerante del motor		OK	
Fugas de aceite y refrigerante		OK	
Estado de cableado eléctrico del motor		OK	
Voltaje de batería y nivel del motor	13.5 V	OK	
Nivel de combustible del tanque		OK	

Compresión de los cilindros	120.3 psi	OK	
-----------------------------	-----------	----	--

Una vez que se ha comprobado la correcta ejecución del motor y culminado el proceso de mantenimiento, se procede a la instrumentación de este para obtener señales del sensor MAP, mediante la aplicación de métodos experimentales al provocar las fallas en el MEP y obtener una base de datos.

2.2.Fallas provocadas

Mediante métodos experimentales se provocaron un total de 19 fallas mecánicas en el MEP, como se muestra en el trabajo de (Leon Japa y Maldonado Ortega), las mismas que se enlistan a continuación en la tabla 5.

Tabla 6 Codificación de fallas a simular.

Fuente: Autor

Descripción de la falla.	Código
Falla del Cilindro 1 desconectado el Inyector.	100
Falla del Cilindro 2 desconectado el inyector.	200
Falla del Cilindro 3 desconectado el Inyector.	300
Falla del Cilindro 4 desconectado el inyector.	400
Falla del Cilindro 1 calibración de la bujía a cero.	500
Falla del Cilindro 2 calibración de la bujía a cero.	600
Falla del Cilindro 3 calibración de la bujía a cero.	700
Falla del Cilindro 4 calibración de la bujía a cero.	800
Desconexión de bobina, falla de cilindros 1 y 3.	900
Desconexión de bobina, falla de cilindros 2 y 4.	1000
Falla en la distribución, adelantar un diente en el árbol de levas de la admisión.	1100
Falla en la distribución, adelantar un diente en el árbol de levas del escape.	1200
Falla en distribución, retrasar un diente en el árbol de levas de admisión.	1300
Falla en la distribución, retrasar un diente en el árbol de levas de escape.	1400
Falla en la distribución, retrasar un diente en el árbol de levas de escape y admisión.	1500
Falla en la distribución, adelantar un diente en el árbol de levas de escape y admisión.	1600
Filtro de aire tapado a un 50%.	1700
Filtro de aire tapado a un 75%.	1800
Motor no presenta falla.	0

2.2.1. Fallas provocadas de un cilindro al desconectar el inyector

La desconexión de los inyectores se genera mediante el panel de control de la maqueta como se observa en la figura 19. Al accionar el interruptor, el inyector suspende la inyección

de combustible en la cámara del cilindro, por lo que no se produce combustión en este, lo que genera una falla en el funcionamiento del cilindro.

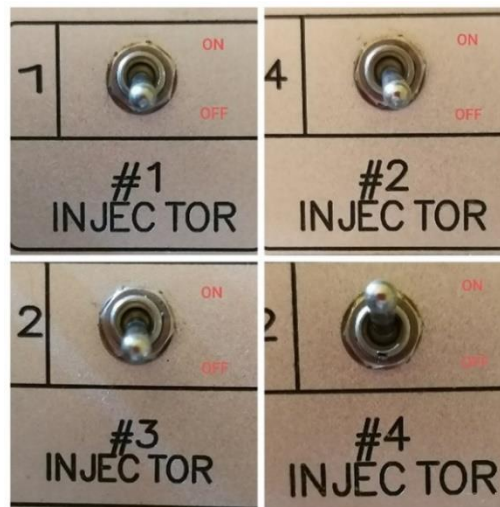


Figura 19 Interruptores de los inyectores.

Fuente: Autor

Cuando se desconecta el primer cilindro, es necesario esperar 5 minutos para que el motor se estabilice y obtenga la señal del sensor. Este procedimiento se repite para todos los inyectores. La falla que fue provocada corta la alimentación de combustible al inyector, debido a que el proceso de llenado de la cámara de combustión no se efectúa. Esta falla se debe a varios factores, lo cual puede ser ocasionado por un cortocircuito, mal contacto en el socket, pistas o drivers quemados.

2.2.2. Fallas provocadas ajustando la apertura del electrodo de la bujía

En esta falla la bujía se encuentra totalmente cerrada, como se muestra en la sección a de la figura 20, por lo que no se producirá la chispa para la combustión en el cilindro, al estar totalmente unido el electrodo central con el electrodo masa.

La bujía del cilindro calibrada a 1 mm, como se muestra en la sección b de la figura 20, es cambiada por la bujía calibrada a medida de fabricante. El resto de los cilindros se mantiene con las medidas del fabricante. Este proceso se repite para todos los cilindros para obtener los datos de simulación de la falla.

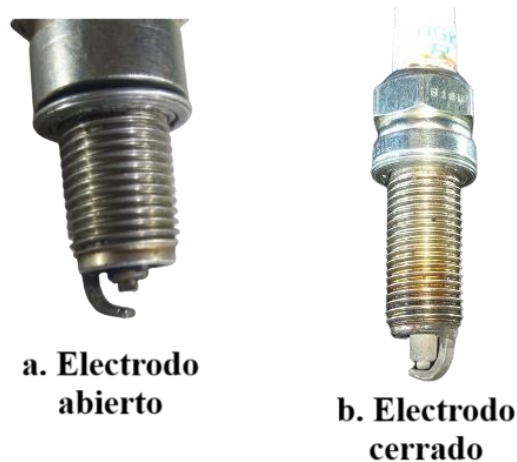


Figura 20 Calibración de bujías.

Fuente: Autor

2.2.3. Fallas provocadas en las bobinas de encendido

La bobina es encargada de recibir la energía desde la batería en voltios y convertirla en una tensión de kilovoltios por especificaciones técnicas del banco didáctico. Debido a que se cuenta únicamente con 2 bobinas, se induce la falla al mismo tiempo en 2 cilindros. Primero se desactivó la bobina 1-4 luego se desactivó la bobina 2-3 cómo se observa en las figuras 21 y 22.

Esta falla se puede presentar en el motor en forma de bobina quemada, o un corte de corriente en el riel de cables. Mediante la señal del sensor MAP, se creó una base de datos con la recolección de información de la falla simulada.



Figura 21 Interruptor de bobina #2 #3.

Fuente: Autor



Figura 22 Interruptor de bobina #1 #4.

Fuente: Autor

2.2.4. Fallas provocadas en sistema de distribución

El banco didáctico cuenta con motor Hyundai Sonata 2.0, el cual posee 2 árboles de levas, uno de admisión y uno de escape. Para su correcto armado y funcionamiento, el cabezote, los árboles de levas y poleas cuentan con cuatro marcas de referencia. Estas marcas se ubican en la polea del árbol de levas de admisión, en la polea del árbol de levas de escape, en la polea del cigüeñal y en el centro de masa del cigüeñal. Para iniciar con la recolección de datos, estas marcas deben estar todas perfectamente ubicadas y ajustadas.

Intencionalmente se realiza un proceso de armado de la distribución distinto del fabricante, al tomar como punto de referencia un diente de la polea e inducir a que este se retrase o adelante según sea el caso necesario, como se observa ver en la figura 23.

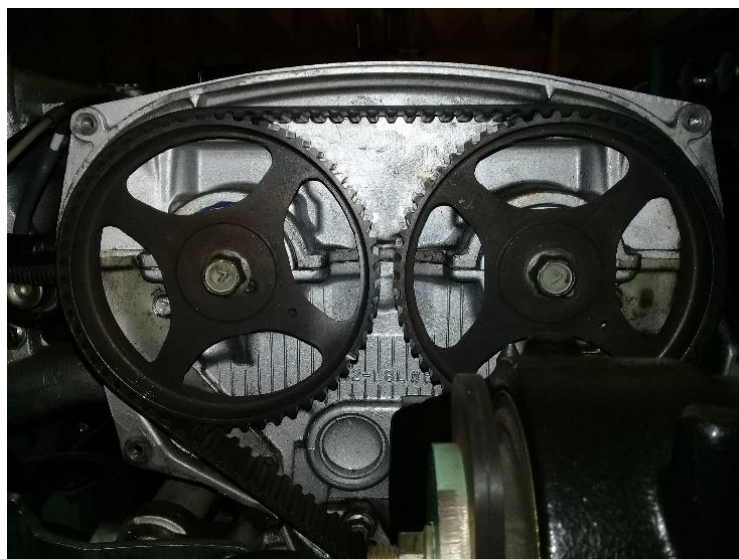


Figura 23 Calibración de distribución.

Fuente: Autor

2.2.5. Fallas provocadas al filtro de aire.

El motor debe tener el filtro de aire en óptimas condiciones, debido a que su trabajo es impedir que la suciedad, los residuos e impurezas lleguen a la cámara de combustión y puedan generar daños en el motor. Un filtro de aire sucio significa una reducción en las prestaciones del vehículo, por esto para inducir esta falla se tapa el filtro de aire en un 50\% y 75\% de su superficie cómo se muestra en la figura 21.



Figura 24 Filtro de aire tapado al 50 %.

Fuente: Autor



Figura 25 Filtro tapado al 75%.

Fuente: Autor

2.2.6. Factores no controlados del experimento

Existen variables que no son posibles de controlar durante el desarrollo del proyecto. Los valores de las variables no controlables son aplicables para la ciudad de Cuenca la cual se encuentra a 2550 m.s.n.m sobre el nivel del mar, las variables son:

- La presión atmosférica es de 74.21 kilopascales.
- La humedad relativa oscila entre un 41% y 83% anual, obteniendo un valor medio de 62.87%.

2.3.Toma de muestras, creación de base de datos

Para obtener la señal del sensor MAP, se enlazan los pines de la señal junto con el pin negativo del sensor a la Raspberry pi, además se agregó una resistencia de 10 k ohm en el pin de conexión del sensor.

2.3.1. Conexión y configuración de la raspberry pi

Para un correcto funcionamiento de la raspberry debe cumplir algunos requerimientos básicos:

- Conectar la raspberry a una fuente de poder de 5.1 V.
- Conectar una pantalla, teclado y ratón, este requerimiento es único para la función de configuración.
- Instalar el sistema operativo Raspian.

2.3.2. Caracterización del sensor MAP

La caracterización del sensor MAP es necesaria para comprender su comportamiento y reacción ante los estímulos y cambios dentro del entorno. Para establecer un comportamiento se determina la calidad y confiabilidad de las mediciones, mediante variaciones de presión a la que se somete la membrana interna del sensor para medir el voltaje de salida. Los datos se presentan en la tabla 7.

La ecuación característica del sensor MAP es, $4.3649 x^2 + 11.4419 x - 0.42399$ y su grafica de curva característica se ve en la figura 26.

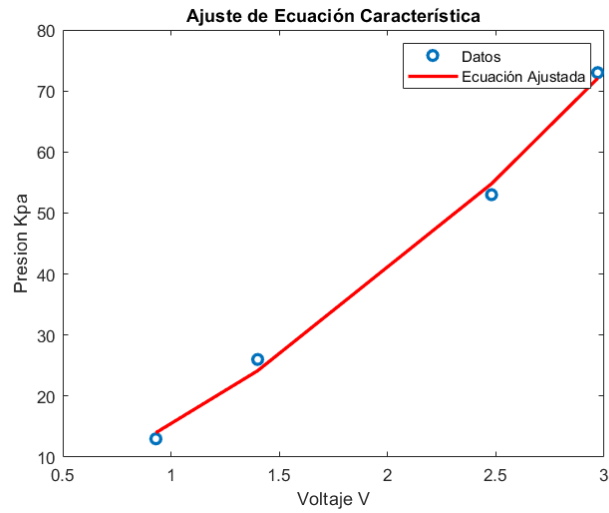


Figura 26 Curva característica del sensor.

Fuente: Autor

Tabla 7 Caracterización sensor MAP.

Fuente: Autor

Presión (kpa)	Voltaje (V)
13	0.93
26	1.4
53	2.48
73	2.97

2.3.3. Esquema de conexión sensor – raspberry

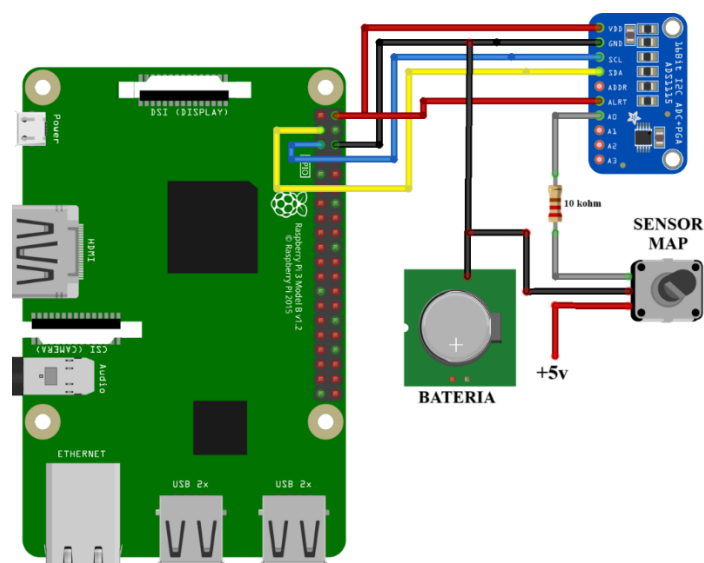


Figura 27 Esquema de conexión sensor - raspberry.

Fuente: Autor

En tabla 8 se muestra las características del cableado del sensor y en la figura 27, se muestra el esquema de conexión del sensor MAP, siendo este representado por un potenciómetro, en la figura 28 se muestra el Broadcom de la Raspberry. Para esta conexión exitosa se deben seguir los siguientes pasos:

- El pin 2 de la broadcom es el pin de alimentación de 5 V y se conecta al pin VDD del ADS1115.
- El pin 6 de la broadcom se conecta al pin GND del ADS1115
- El pin 3 de la broadcom se conecta al pin SDA del ADS1115
- El pin 5 de la broadcom se conecta al pin SCL del ADS1115.
- El pin ADDR del ADS1115 se conecta con un empalme al cable de alimentación de 5v.
- El pin A0 del ADS1115 se conecta a un extremo de la resistencia mientras que el otro extremo se conecta al sensor MAP en su pin de señal.
- A través de un empalme al cable del pin de GND, este se conecta al negativo de la batería.

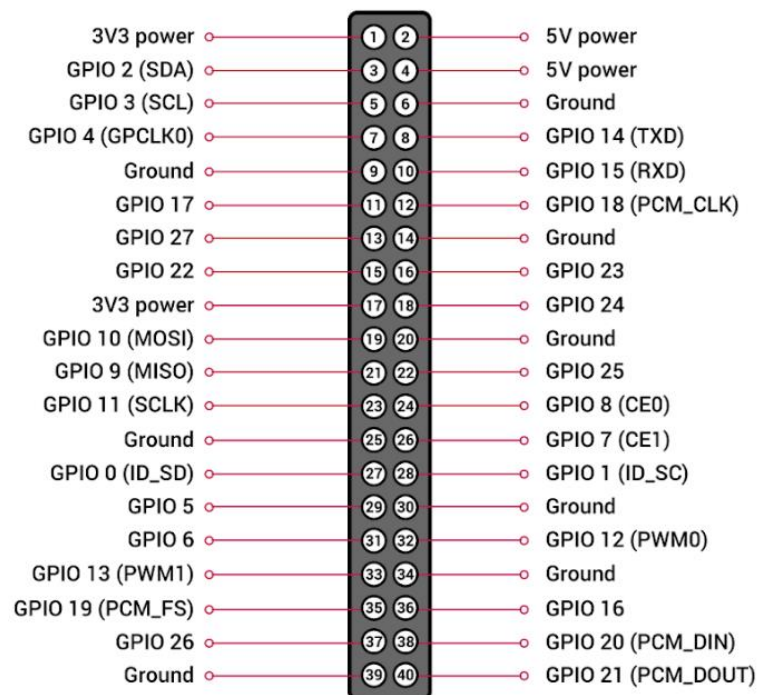


Figura 28 Broadcom de los pines GPIO de la raspberry.

Fuente: (solectroshop)

Tabla 8 Identificación del cableado del sensor MAP.

Fuente: Autor

Características sensor MAP	
Color	Función
Negro	Tierra
Azul	Señal sensor IAT
Verde	Señal sensor MAP
Rojo	Alimentación

2.3.4. Protocolo para la adquisición de datos para la creación de base de datos.

Una vez establecidas las conexiones del sensor MAP se procede a la adquisición de las señales mediante un algoritmo desarrollado que obtiene y guarda los valores de la señal temporal. A continuación, en la figura 29, se explica el proceso de adquisición a través de un flujograma.

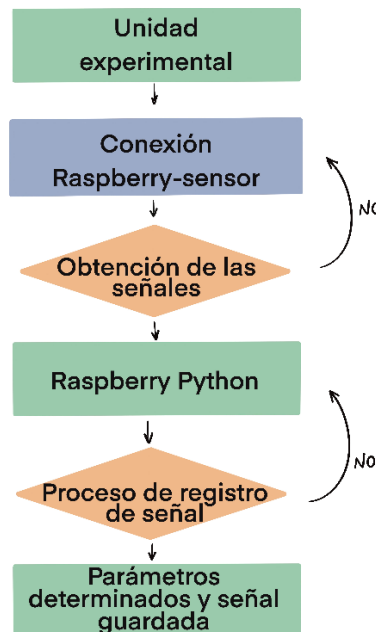


Figura 29 Flujograma para la obtención de señales.

Fuente: Autor

Se determina que el tiempo de muestreo optimo es de 3 minutos, según el trabajo de (Leon Japa y Maldonado Ortega), es importante que el motor trabaje a su temperatura optima de funcionamiento, para una correcta toma de datos. También es primordial considerar que, al

momento de realizar la toma de datos, no deben funcionar los elementos consumidores de energía del motor, como el electroventilador.

Esta toma de datos posee los valores del sensor MAP junto con la fecha y hora de la lectura del dato. Son alrededor de 5000 datos por cada muestra. Por cada uno de los 18 fallos mecánicos simulados se tomaron 12 muestras, lo que da una base de datos para entrenamiento de 60000 datos.

La recolección de datos se efectúa mediante un algoritmo desarrollado en el software Python, el cual lee y guarda los valores de la señal del sensor. Después determina los parámetros estadísticos del conjunto de datos, con una frecuencia de muestreo de unos 5000 datos por 180 segundos, este algoritmo se observa en el anexo #2.

Trascurrido este tiempo, los datos de la señal del sensor y sus atributos estadísticos se almacenan en un archivo tipo texto (txt) para su procesamiento, a continuación, se enumeran los pasos para lograr una adecuada recolección de datos.

1. La Raspberry Pi se conecta a una fuente de poder. En el caso de ser la toma de datos inicial, los elementos secundarios como pantalla, teclado y ratón son conectados. Una vez comprobadas las conexiones previas se enciende la Raspberry.
2. La Raspberry cargada con el sistema operativo Raspbian, disponible en la página web de Raspberry.
3. Se identifican los pines del sensor MAP requeridos para la conexión. Los pines a identificar son el pin de señal y el pin negativo del sensor MAP.
4. El pin de señal de voltaje del sensor se conecta al circuito, el cual tiene un pin asignado que es el pin de salida de la resistencia de 10 Kohm. A su vez el pin negativo del sensor se conecta al pin negativo común entre el circuito, la placa, el sensor y la batería.
5. Puesto a punto el hardware y previo a la ejecución del algoritmo, se enciende el motor del banco didáctico por un tiempo de 7 min, para que llegue a su temperatura de funcionamiento.
6. Luego de alcanzar la temperatura de funcionamiento del motor, se ejecuta el algoritmo, y este comienza a recolectar los valores de voltaje del sensor.
7. Concluido el tiempo de recolección de 180 segundos, se determinan los valores de voltaje y atributos estadísticos de la señal, luego esta información se guarda en un archivo (txt) que contiene los valores de voltaje y atributos estadísticos de la señal del sensor, se realiza este proceso por 12 veces para cada falla simulada.

8. Se crea una base de datos, en forma de una matriz, con los valores de los atributos estadísticos, esto se efectúa con la ayuda de una herramienta graficadora desarrollada en LabVIEW, que crea una matriz de datos en un archivo tipo .xlsx, como se ve en la figura 30, con los valores de todas las muestras recolectadas.

Arithmetic	RMS	Standard d	Variance	Kurtosis	Median	Mode	Sumatoria	Skewness	Maximo	Minimo	Range	Total samples	Resultado
1,286	1,286	0,023	0,001	2,109	1,285	1,308	6430,365	0,154	1,349	1,235	0,114	5000	0
1,274	1,274	0,023	0,001	2,119	1,273	1,27	6371,331	0,166	1,339	1,222	0,117	5000	0
1,282	1,282	0,023	0,001	2,17	1,281	1,301	6409,752	0,167	1,352	1,229	0,123	5000	0
1,284	1,284	0,023	0,001	2,151	1,283	1,266	6420,352	0,159	1,349	1,229	0,12	5000	0
1,272	1,272	0,023	0,001	2,092	1,271	1,258	6360,219	0,164	1,334	1,221	0,113	5000	0
1,279	1,279	0,023	0,001	2,161	1,278	1,267	6394,823	0,176	1,354	1,23	0,125	5000	0
1,276	1,276	0,024	0,001	2,246	1,275	1,262	6380,349	0,162	1,348	1,219	0,129	5000	0
1,275	1,276	0,023	0,001	2,177	1,274	1,269	6376,939	0,163	1,348	1,225	0,124	5000	0
1,277	1,277	0,023	0,001	2,203	1,276	1,259	6384,913	0,175	1,346	1,226	0,121	5000	0
1,272	1,272	0,023	0,001	2,139	1,271	1,284	6360,118	0,157	1,336	1,223	0,114	5000	0
1,282	1,282	0,024	0,001	2,191	1,282	1,269	6410,868	0,128	1,353	1,227	0,126	5000	0
1,281	1,282	0,023	0,001	2,097	1,281	1,257	6407,166	0,146	1,343	1,234	0,11	5000	0

Figura 30 Matriz de datos estadísticos.

Fuente: Autor

2.3.4.1. Validación de resultados

El procesamiento, validación de la muestra y creación de la matriz de datos, se efectúa con la ayuda de una herramienta graficadora desarrollada en LabVIEW como se muestra en la figura 31, que genera una matriz en con todos los valores estadísticos recolectados, esta matriz es utilizada para la creación y entrenamiento de la red neuronal.

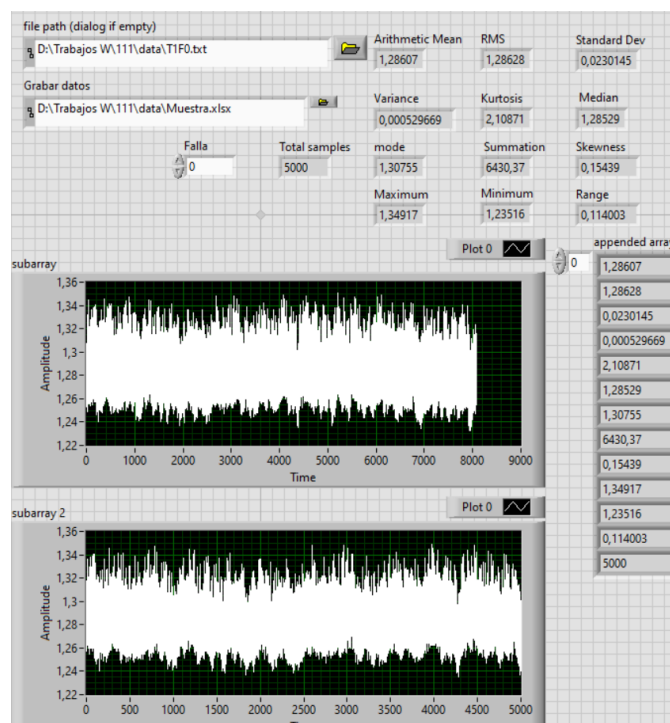


Figura 31 Graficadora LabVIEW.

Fuente: Autor

Anterior al proceso de creación y entrenamiento del algoritmo de clasificación, se realiza un análisis en Matlab para determinar correlación y validez de los datos recolectados, este análisis se practica con la creación de una red neuronal de prueba, que expone un resultado afirmativo con validez de los datos y nos determina un error aproximado del 18%, usando un entrenamiento y procesamiento estándar de las muestras, como se muestra en la figura 32. Esto indica que este valor de error puede disminuir según el método de entrenamiento aplicado.

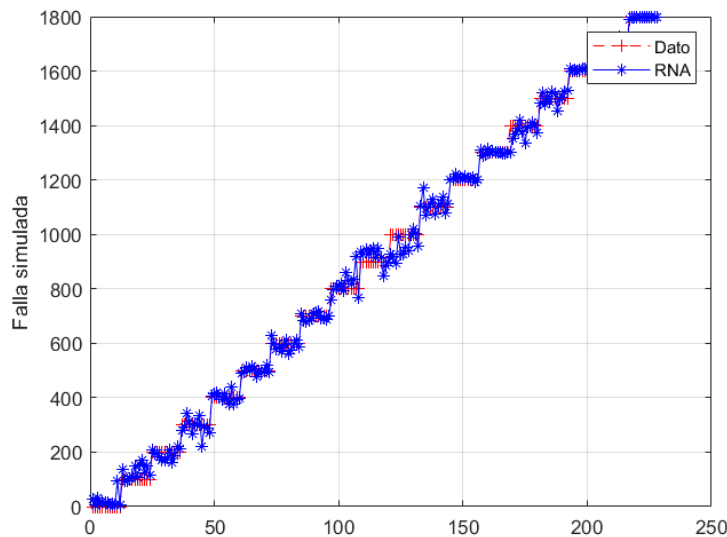


Figura 32 Resultante de error, red de prueba.

Fuente: Autor

2.3.5. Análisis de varianza

Según (Minitab) el método estadístico (ANOVA), comprueba la hipótesis de que las medias de 2 o más poblaciones son iguales.

Se desarrolla el análisis ANOVA de un solo factor para determinar los mejores atributos de entrada que ingresan a la matriz de selección general.

El análisis de Varianza (ANOVA) de un solo factor se aplica a todos los atributos estadísticos para determinar los valores que clasifican los atributos. Algunos de estos son, valor p , gráfica de intervalos, R cuadrada, gráfica de residuos, comparaciones en parejas de Tukey.

- El R cuadrado determina el porcentaje de variación en la respuesta. De tal manera mientras mayor sea el valor de R cuadrado, mejor se ajusta el modelo a los datos recolectados.

- El valor de P determina si los resultados son estadísticamente significativos. El valor de P oscila entre 0 y 1. Para un nivel de significancia, suele usarse un valor de 0.05, entonces, si el valor P es menor que o igual a 0.05, el valor tiende a ser no significativo.

Para el desarrollo del ANOVA de un solo factor, se procedió a insertar los factores estadísticos en la hoja de trabajo del Software estadístico Minitab respectivamente, cada factor en cada columna. Luego, en la columna final se agrega el código de cada falla simulada.

Las gráficas de resultados obtenidas mediante el ANOVA, aplicada a la variable estadística de la moda, se muestra en las figuras 33, 34 y 35. En donde se ilustra las gráficas de residuos intervalos y comparaciones en parejas de tukey, respectivamente.

En la gráfica 33 (Residuos vs Valor ajuste) se verifica que, en la variable moda, los puntos no representan ningún patrón en forma de embudo o corneta debido a que se encuentran distribuidos aleatoriamente, por lo que se acepta el supuesto de una varianza constante.

En la figura 33 (Residuos vs Orden de observación), se verifica que los puntos, son independientes entre sí, ya que no están correlacionados. Se encuentran ubicados aleatoriamente alrededor de la línea central o línea cero.

En la figura 33 (Gráfica de probabilidad normal), se acepta el supuesto de normalidad, ya que se corrobora que los datos están distribuidos normalmente. Los datos siguen la tendencia a una línea recta.

En la figura 33 (Histograma) se muestra una correcta distribución de los datos.

En la figura 34, se verifica la variación e independencia de las muestras de la variable estadística MEDIA, ya que presenta una aleatoriedad.

Para determinar si existe relación en los grupos, se usa el método de tukey, el mismo establece, que los grupos que no comparten información tienen una diferencia media que es estadísticamente significativa.

En la figura 35 de comparaciones en parejas de tukey de la media, se observa que las diferencias de las medias son evidentemente diferentes, por lo que este factor estadístico es correcto usar para la matriz de selección general. Se comprueba estadísticamente que existe la variación de las señales en diferentes condiciones de funcionamiento. Esto provoca concluir que el sistema realiza la acción de diferenciar las fallas mecánicas en el motor.

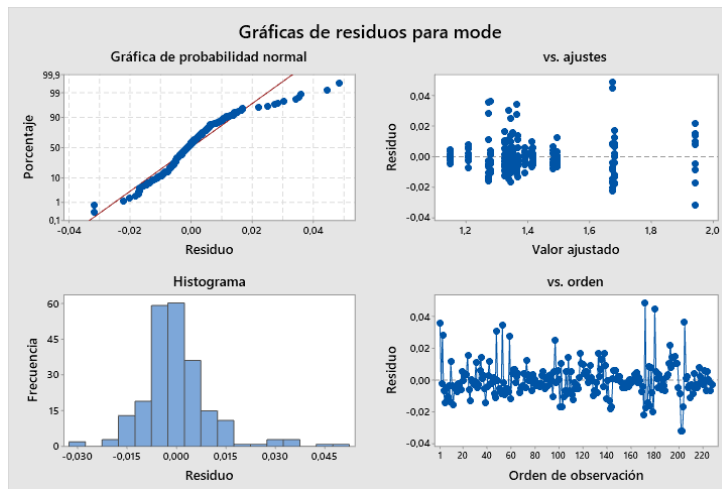


Figura 33 Gráfica de residuos para la variable estadística MODA.

Fuente: Autor

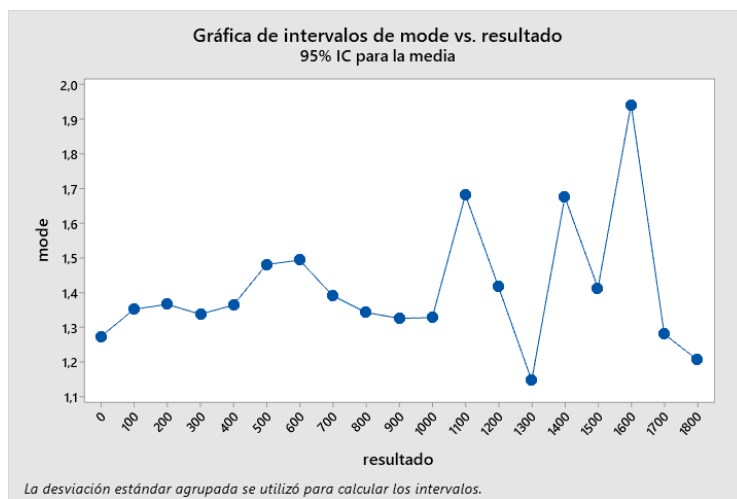


Figura 34 Gráfica de intervalos de MODA vs RESPUESTA.

Fuente: Autor

Comparaciones en parejas de Tukey

Agrupar información utilizando el método de Tukey y una confianza de 95%

resultado	N	Media	Agrupación
1600	12	1,94283	A
1100	12	1,68108	B
1400	12	1,67525	B
600	12	1,49342	C
500	12	1,48067	C
1200	12	1,41600	D
1500	12	1,41283	D
700	12	1,39033	E
200	12	1,36600	F
400	12	1,36350	F
100	12	1,35133	F G
800	12	1,34275	G H
300	12	1,33742	G H I
1000	12	1,32767	H I
900	12	1,32542	I
1700	12	1,27983	J
0	12	1,27250	J
1800	12	1,20592	K
1300	12	1,14642	L

Las medias que no comparten una letra son significativamente diferentes.

Figura 35 Comparación de parejas de Tukey de la variable estadística MODA.

Fuente: Autor

2.4. Diseño del sistema inteligente de diagnóstico en plataforma Android.

En base a una previa revisión del Estado del arte, se determina que la herramienta Random Forest Classifier, cumple con los requisitos de comunicación, entrenamiento, requerimientos necesarios para el desarrollo completo del sistema inteligente de diagnóstico.

2.4.1. Creación y entrenamiento del algoritmo de predicción.

El entrenamiento y creación del algoritmo se explica en el flujograma representado en la figura 36, se escribe un código de programación que se puede ver en el anexo #2.

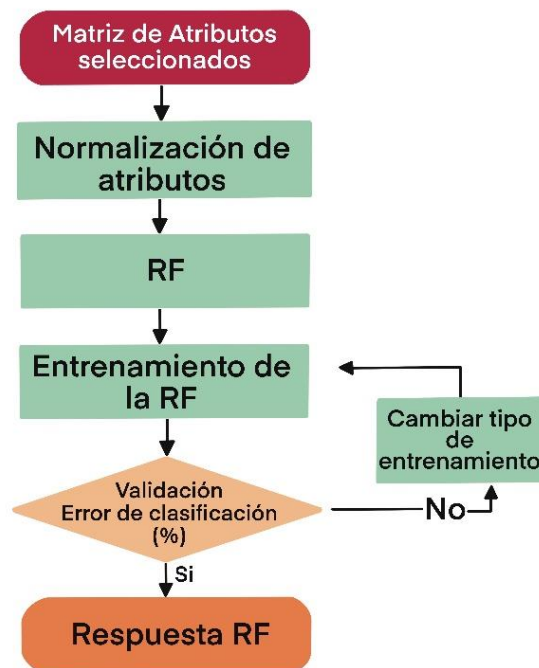


Figura 36 Flujograma de creación y entrenamiento del algoritmo.

Fuente: Autor

Los pasos para obtener un archivo entrenado que pueda utilizarse para hacer predicciones son:

1. Obtención de Datos:

Obtener una base de datos etiquetados con ejemplos de características y sus respectivas etiquetas de clase.

2. División de Datos:

Como se muestra en la figura 37, se dividen los datos, en conjuntos de entrenamiento y prueba. El conjunto de entrenamiento se utiliza para enseñar al modelo, y el conjunto de prueba evalúa su rendimiento.

```
1 # Importación de bibliotecas necesarias
2 import pandas as pd
3 from sklearn.model_selection import train_test_split
4 from sklearn.ensemble import RandomForestClassifier
5 from sklearn.metrics import accuracy_score, classification_report
6 import matplotlib.pyplot as plt
7 from joblib import dump, load
8
9 # Lectura de datos desde un archivo Excel
10 df = pd.read_excel('base de datos.xlsx')
11
12 # Preparación de datos
13 X = df.drop(['total samples', 'resultado'], axis=1) # Variables independientes
14 y = df['resultado'] # Variable dependiente
15
16 # División de datos en conjuntos de entrenamiento y prueba
17 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Figura 37 Fragmento 1 de código, del algoritmo de entrenamiento.

Fuente: Autor

3. Configuración de Parámetros:

En base a una previa revisión del Estado del arte, se establece que para este caso los parámetros se determinan en los valores predeterminados del algoritmo, que serían:

- Para el número máximo de características a considerar para dividir un nodo, su valor predeterminado en "automático", equivale a la raíz cuadrada del número total de características.
- Para el número de árboles en el bosque, su valor predeterminado es 100 árboles.
- Para la profundidad máxima de cada árbol: los árboles se expandirán hasta que todas las hojas cumplan con contener menos que el hiperparámetro "min_samples_split", el cual determina el número mínimo de muestras que deben estar presentes en un nodo para que se considere la posibilidad de dividir ese nodo en dos nodos hijos durante la construcción del árbol.
- Para el número mínimo de muestras requeridas para dividir un nodo, el valor predeterminado es 2, ya que se requieren al menos 2 muestras en un nodo para que se pueda dividir.

- Para el número mínimo de muestras requeridas en una hoja el valor predeterminado es 1, ya que se requiere al menos 1 muestra en cada hoja del árbol.

4. Entrenamiento del Modelo:

Como se observa en la figura 38 durante este proceso, el algoritmo crea árboles de decisión, cada uno entrenado con un subconjunto aleatorio de datos.

```
1 # Creación del modelo de clasificación RandomForest
2 classifier = RandomForestClassifier(random_state=42)
3
4 # Entrenamiento del modelo con los datos de entrenamiento
5 classifier.fit(X_train, y_train)
6
7 # Guardar el modelo entrenado en un archivo
8 model_filename = 'RandomForestClassifier.joblib'
9 dump(classifier, model_filename)
10 print(f"Modelo de Clasificación guardado como '{model_filename}')
```

Figura 38 Fragmento 2 de código, del algoritmo de entrenamiento.

Fuente: Autor

5. Guardado del Modelo:

Una vez que el modelo está entrenado, se crea y guarda un archivo o una estructura de datos específica del lenguaje de programación, este archivo será la AI, lista para su aplicación.

2.4.2. Diseño del protocolo de comunicación del sistema de predicción.

Desarrollados los algoritmos de recolección de datos del sensor y predicción de fallas. Se crea un protocolo de comunicación entre los algoritmos, la Raspberry y la aplicación Android.

Esta interconexión se dará mediante el Protocolo de Comunicación MQTT, con la ayuda de la herramienta web conocida como “HIVEMQ”, la interfaz de esta herramienta se muestra en la figura 39, la herramienta se agrega a Python para su uso, en forma de biblioteca. Esta herramienta tiene la capacidad de comunicación y ejecución desde la Raspberry, creando un nodo de comunicación en la nube, como se observa en el diagrama de la figura 40.

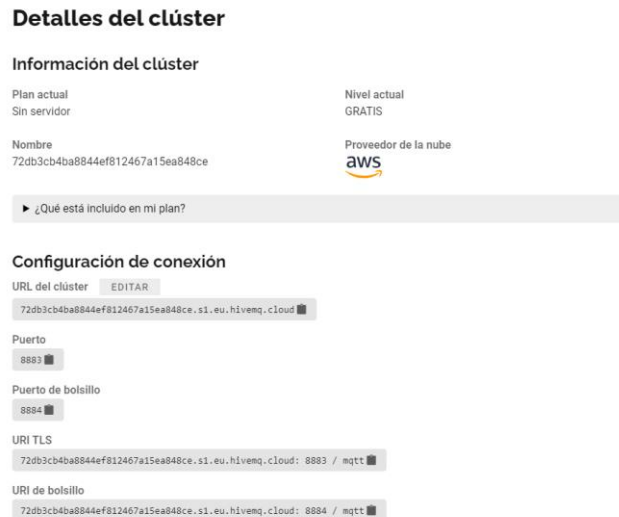


Figura 39 Interfaz Hive MQ.

Fuente: Autor

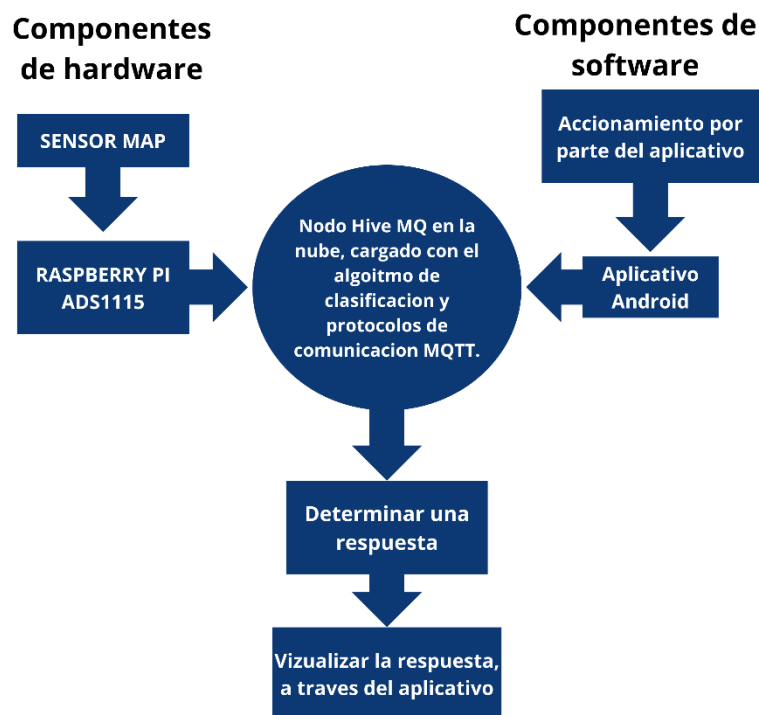


Figura 40 Diagrama de funcionamiento de la comunicación del sistema.

Fuente: Autor

Como se muestra en las figuras 41 y 42, a este nodo se carga, el algoritmo de adquisición de datos del sensor y el algoritmo de predicción de fallas. Luego se configura la comunicación, para el control del sistema y visualización de resultados, que se realiza mediante la aplicación desarrollada en Android.

```

1 # Comunicacion
2 client = paho.Client(client_id="", userdata=None, protocol=paho.MQTTv5)
3 client.on_connect = on_connect
4 # habilite TLS para una conexión segura
5 client.tls_set(tls_version=mqtt.client.ssl.PROTOCOL_TLS)
6 # establecer nombre de usuario y contraseña
7 client.username_pw_set("modelo_ia", "modeloia2024")
8 # conectarse a HiveMQ Cloud en el puerto 8883 (predeterminado para MQTT)
9 client.connect("72db3cb4ba8844ef812467a15ea848ce.s1.eu.hivemq.cloud", 8883)
10 # configurando devoluciones de llamada, use funciones separadas como las anteriores para una mejor visibilidad
11 client.on_subscribe = on_subscribe
12 client.on_message = on_message
13 client.on_publish = on_publish
14 # suscríbese a todos los temas de la enciclopedia utilizando el comodín "#"
15 client.subscribe("Rasp/Raspi/#", qos=1)
16 client.loop_start()

```

Figura 41 Fragmento 3 de código, del algoritmo de entrenamiento.

Fuente: Autor

```

1 public String ConectarHiveMQ() {
2     try {
3         // Intenta conectar al servidor HiveMQ
4         Mqtt5ConnAck connAckMessage = client.toBlocking().connect();
5         // Éxito
6         return connAckMessage.getReasonCode().toString();
7     } catch (Exception e) {
8         // Fallo
9         return e.getMessage();
10    }
11 }
12 #Este método intenta conectar al servidor HiveMQ de forma bloqueante (client.toBlocking().connect()),
13 #y devuelve el código de razón de la conexión (connAckMessage.getReasonCode().toString()).
14 #Si hay algún fallo durante la conexión, se captura la excepción y se devuelve el mensaje de error.

```

Figura 42 Fragmento 4 de código, del algoritmo de entrenamiento.

Fuente: Autor

Se utilizó el algoritmo Random Forest Classifier para predecir resultados en función de un conjunto de datos de entrada. Después de realizar la predicción, se calcula el error comparando las predicciones del modelo con los valores reales, y la matriz de confusión para obtener información detallada sobre los resultados de la clasificación, la matriz de confusión compara las predicciones del modelo con los valores reales y cuenta la cantidad de verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos, se puede apreciar el fragmento de código de cálculo del error en la figura 43.

```

1 from sklearn.metrics import confusion_matrix, precision_score, recall_score, f1_score
2 # Realizo la predicción con el modelo cargado
3 predicted_result = classifier_loaded.predict(X_test)
4 # Calculo la matriz de confusión
5 conf_matrix = confusion_matrix(y_test, predicted_result)
6 # Calculo otras métricas
7 precision = precision_score(y_test, predicted_result, average='weighted')
8 recall = recall_score(y_test, predicted_result, average='weighted')
9 f1 = f1_score(y_test, predicted_result, average='weighted')
10 # Imprimo los resultados
11 print("Matriz de Confusión:")
12 print(conf_matrix)
13 print("\nPrecision:", precision)
14 print("Recall:", recall)
15 print("F1-score:", f1)

```

Figura 43 Fragmento 5, cálculo del error.

Fuente: Autor

2.4.3. Diseño del aplicativo Android.

El aplicativo móvil desarrollado, para que el usuario manipule la información de manera intuitiva, se condensa todo en una interfaz o pantalla, como se observa en la figura 44.



Figura 44 Interfaz de usuario en el aplicativo Android.

Fuente: Autor

Es imprescindible que se especifiquen las características que debe desempeñar la aplicación, algunas de ellas son:

- Didáctica e interactiva para los usuarios
- La aplicación desarrollada bajo el sistema operativo Android
- Su disponibilidad debe ser ilimitada
- La aplicación permitirá el control del inicio y finalización de los procesos del sistema de diagnóstico
- La aplicación permitirá la visualización de los resultados del sistema de diagnóstico
- Descarga gratuita para el usuario
- Compatible con el protocolo de comunicación HQT

2.4.3.1. Diseño y uso de la aplicación

- Para la ejecución del aplicativo, como se ve en la figura 45, la primera acción, es confirmar la conexión de los elementos, al presionar el botón CONECTAR, se ejecutan los códigos de comunicación, y emite el mensaje de CONEXIÓN EXITOSA.

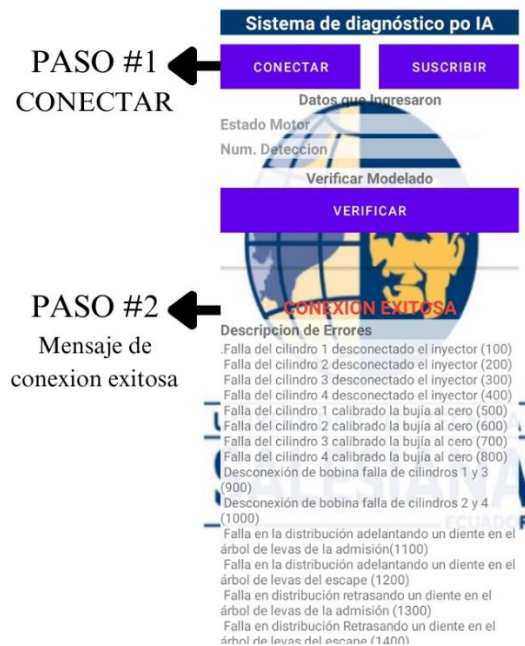


Figura 45 Explicación de funcionamiento del sistema, pasos 1 y 2.

Fuente: Autor

- Una vez se comprueba la conexión, el sistema está listo para funcionar y empezar a clasificar.

- El proceso de clasificación se inicia al seleccionar el botón VERIFICAR, como se observa en la figura 46.



Figura 46 Explicación de funcionamiento del sistema, paso 3.

Fuente: Autor

- Como se muestra en la figura 47, el progreso de la predicción se observa a través de la barra de progreso ubicada debajo del botón VERIFICAR.

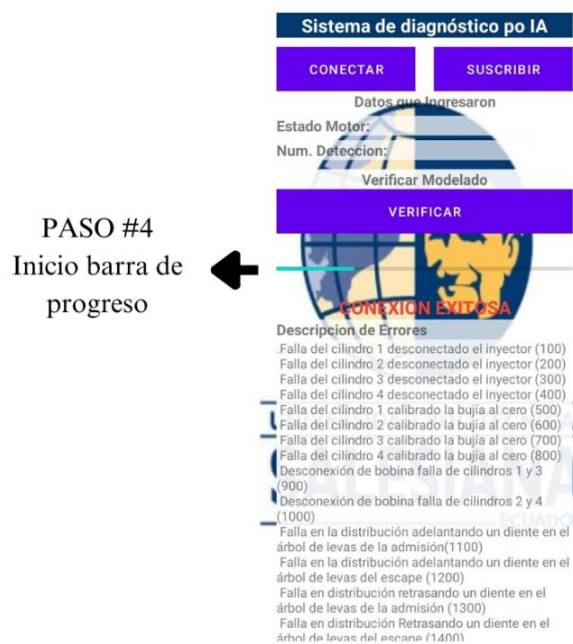
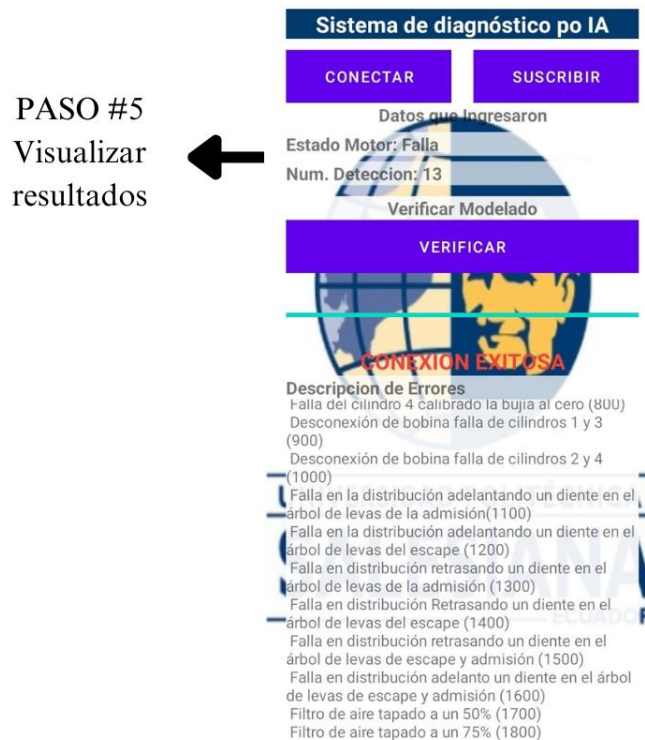


Figura 47 Explicación de funcionamiento del sistema, paso 4.

Fuente: Autor

- Concluido el proceso de clasificación, el resultado se visualiza en los cuadros de texto de ESTADO MOTOR y NUM DETECCION, como se muestra en la figura 48.



PASO #5
Visualizar
resultados

Figura 48 Explicación de funcionamiento del sistema, paso 5.

Fuente: Autor

2.5. Funcionamiento del Sistema inteligente de diagnóstico

Culminado el protocolo de conexión de hardware, se ejecuta el aplicativo Android, donde se visualiza la interfaz del sistema de diagnóstico. Para poner en marcha el sistema, se procede a dar clic en el botón conectar, posteriormente aparece un mensaje que confirma la conexión exitosa, luego se ejecuta el algoritmo al presionar el botón verificar, consecuentemente el algoritmo realiza el análisis de la señal y la obtención de atributos. Finalmente, el algoritmo evalúa la matriz de entrada y muestra el diagnóstico del motor. Este proceso de entrenamiento y generación del algoritmo se explica a través del flujograma de la figura 49.

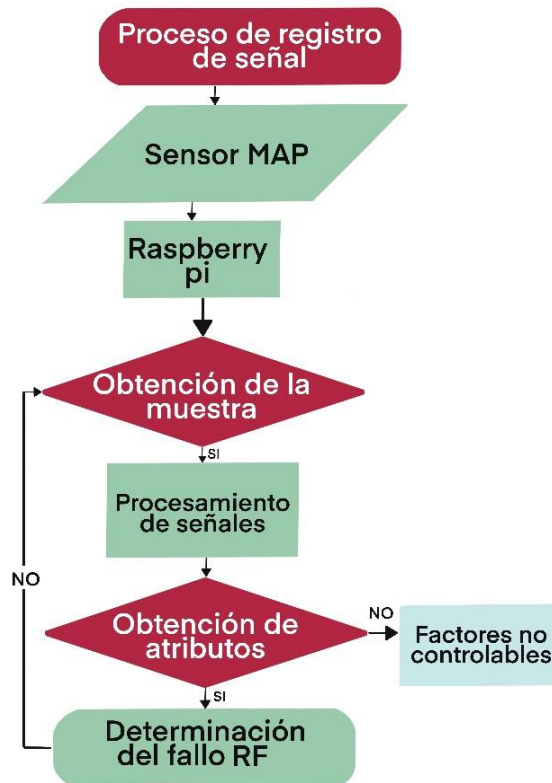


Figura 49 Diagrama de funcionamiento del algoritmo.

Fuente: Autor

3. CAPITULO III: Análisis de resultados

3.1.Resultado de clasificación mediante algoritmo de bosque aleatorio

El algoritmo computacional de Random forest, es eficiente para el sistema inteligente de diagnóstico de fallas mecánicas para motores de encendido provocado, ya que diferencia las anomalías mecánicas provocadas en la unidad experimental. Además, no presenta un gasto computacional alto, por lo que es apto para el caso de estudio.

3.2.Resultados del sistema de diagnóstico inteligente de fallas mecánicas

Se llevan a cabo 12 pruebas en total para verificar la precisión del algoritmo de diagnóstico al introducir una señal temporal que no está presente en la base de datos del algoritmo de clasificación. Estas pruebas implican simular 4 fallas aleatorias distintas en la unidad experimental. Los resultados obtenidos de las 12 pruebas correspondiente a la falla de código 1300, se presentan a detalle en la tabla 9.

Tabla 9 Resultados del motor en falla 1300.

Fuente: Autor

Iteración	Código de falla inducida en el motor.	Código de Falla identificada
1	1300	1300
2	1300	1300
3	1300	1300
4	1300	1300
5	1300	1300
6	1300	1300
7	1300	1300
8	1300	1300
9	1300	1300
10	1300	1300
11	1300	1300
12	1300	1300

Para evaluar el sistema de diagnóstico en diferentes motores de encendido provocado, se verifica que tengan iguales características que la unidad experimental, con la finalidad de comprobar su estado de funcionamiento.

La comprobación se realiza en un segundo motor, en el cual previo a la generación de fallas, se realiza un minucioso proceso de mantenimiento.

Los resultados presentados por el algoritmo de diagnóstico en la detección de fallas mecánicas tienen alta precisión por esta razón, se descarta un error considerable del algoritmo, y una falta de entrenamiento de la red, debido a que los resultados tienen una gran similitud con la falla generada.

3.3. Validación del sistema de diagnóstico

3.3.1. Validación de los datos muestrales

Se realiza un análisis estadístico de la muestra de datos tomada, de cada una de las fallas generadas y se comprueba:

- La probabilidad de que exista normalidad en las muestras obtenidas.
- El supuesto de varianza constante
- La independencia de residuos

Con un análisis para corroborar el efecto de los datos en función a los parámetros de salida. Como se ve en las figuras 50 y 51, en las gráficas de los residuos para los atributos de mediana y moda, existe normalidad en los datos obtenidos, esto se corrobora con la gráfica de histograma de residuos que indica que existe una normalidad simétrica, a la vez en las gráficas

se puede ver que los valores predichos contra los residuos se distribuyen de manera aleatoria sin seguir ningún patrón, debido a que la gráfica no sigue patrones, se acepta el supuesto de varianza constante, como también que los residuos no están correlacionados entre sí y que las muestras son aleatorias.

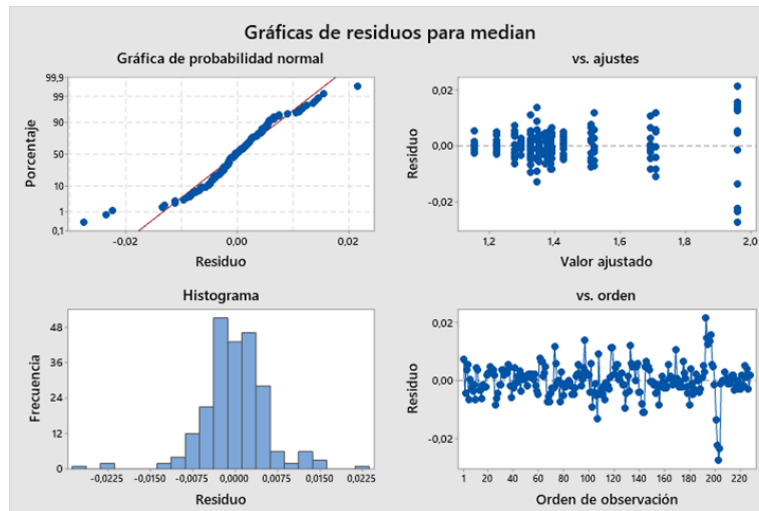


Figura 50 Grafica de residuos de la Mediana.

Fuente: Autor

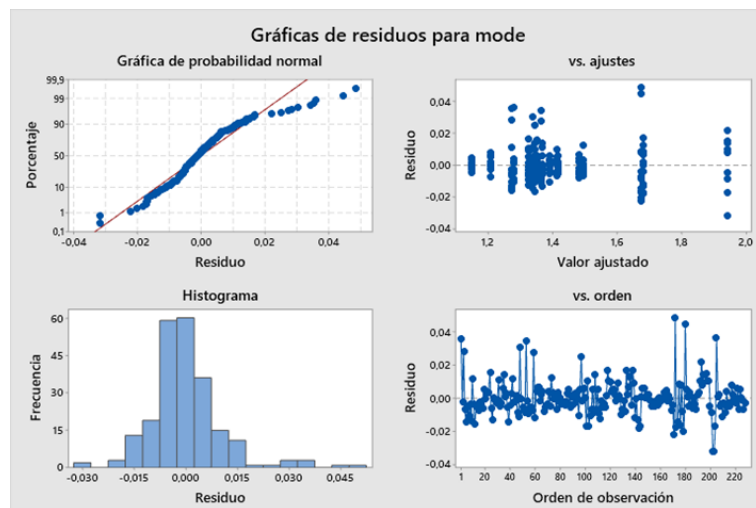


Figura 51 Grafica de residuos de la Moda.

Fuente: Autor

3.3.2. Validación de resultados en la clasificación de las fallas mecánicas

Para validar la clasificación de las fallas es necesario comparar la respuesta real con el resultado de la red utilizada, en el presente sistema de identificación de patrones de fallas se

realiza un análisis de varianza como se muestra en la figura 52, para comprobar la validez de los resultados.

En la figura 53, se visualiza el resultado de los valores de p y r cuadrada, por lo cual se determina que las medias de la respuesta real y del algoritmo de clasificación tienen una equivalencia, el resultado del ANOVA arroja un valor de p de 0.001, y el valor de R cuadrado que se muestra en la figura 53 explica una gran proporción de la variabilidad en la variable dependiente, entre las respuestas. Por lo cual se demuestra que la identificación de fallas mecánicas mediante algoritmo Random forest es eficiente. Por lo que deducimos que el sistema de diagnóstico inteligente posee una confiabilidad del 98.5% .

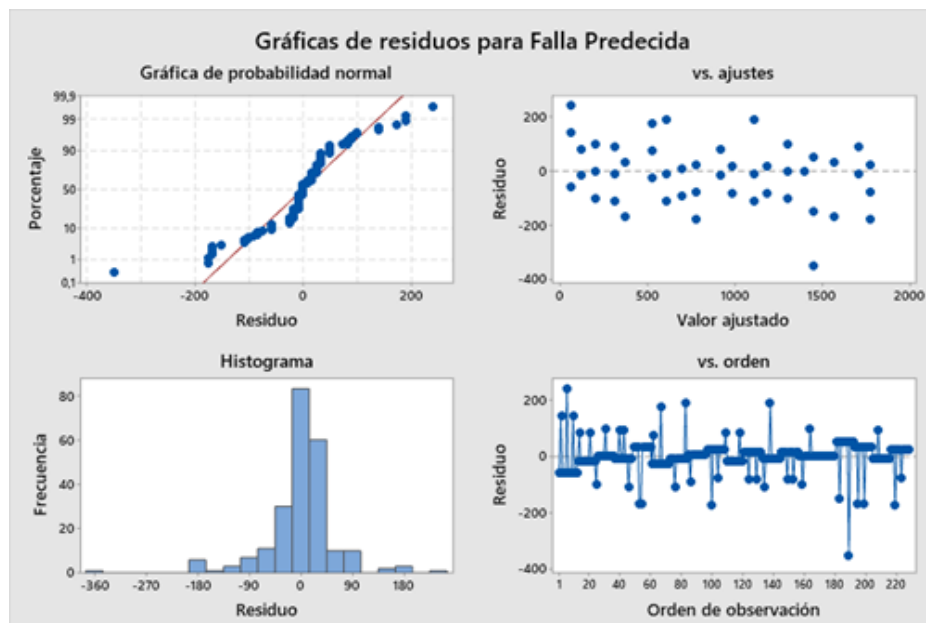


Figura 52 Graficas de residuos del sistema de diagnóstico.

Fuente: Autor

Análisis de Varianza

Fuente	GL	SC Ajust.	MC Ajust.	Valor F	Valor p
Falla Real	18	65255351	3625297	932,54	0,000
Error	209	812500	3888		
Total	227	66067851			

Resumen del modelo

S	R-cuad.	R-cuad. (ajustado)	R-cuad. (pred)
62,3503	98,77%	98,66%	98,54%

Figura 53 Valor de p y r cuadrada.

Fuente: Autor

3.4. Conclusiones

Como principal resultado se desarrolla un sistema inteligente de diagnóstico de fallas mecánicas en motores de encendido provocado a través del sensor MAP, para dispositivos Android, que cumple con el objetivo planteado; El sistema de diagnóstico es controlado por el aplicativo móvil Android, no es invasivo para el motor en la recolección de datos, y es aplicable para diferentes condiciones de funcionamiento de un motor Hyundai Sonata 2.0 debido a que tiene la capacidad de identificar fallas mecánicas a través del procesamiento momentáneo de la señal de voltaje del sensor MAP, y capaz de mostrar los resultados del procesamiento a través del mismo aplicativo.

De la revisión del estado del arte se obtuvo información sobre las variables representativas y condiciones de funcionamiento de los motores MEP y el sensor MAP, se obtienen 11 atributos estadísticos los cuales son; mínimo, máximo, media, mediana, moda, rango, asimetría, suma, curtosis, desviación estándar, media cuadrática. Se desarrolló un sistema de recolección de datos a través del trabajo conjunto de un convertidor analógico digital ADS1115 y una Raspberry Pi 4B con sistema operativo Raspian; con el sistema de recolección de datos operando se desarrolló una base de datos con los atributos estadísticos de la señal recolectada, para la formación y entrenamiento de un algoritmo de predicción de fallas mecánicas con base al algoritmo Random Forest de Machine Learning.

Respecto al análisis de resultados, se determinó que la respuesta real de clasificación es equivalente al resultado obtenido mediante el algoritmo, un valor de p igual a 0,000 de manera que se muestre que no existe una diferencia estadísticamente significativa. En consecuencia, se valida el error de clasificación de aproximadamente el 1.5%.

Para determinar los atributos de estadísticos se aplicó un proceso de selección en función a las capacidades de programación de Python, las propiedades del algoritmo, y la información obtenida en la revisión del estado del arte; luego, a través de un análisis de estadístico de varianza se validó la selección de los atributos. Mediante la construcción del sistema de diagnóstico inteligente y basados en la síntesis del método inductivo se determinó que el algoritmo Random Forest es una herramienta que puede ayudar al diagnóstico automotriz debido a que el uso de esta tecnología permite un diagnóstico en un tiempo de 2 minutos, al usar un procedimiento poco invasivo.

3.5. Recomendaciones

Se recomienda seguir con la investigación sobre el desarrollo de un sistema de diagnóstico inteligente a través de la señal del sensor MAP, aplicando el algoritmo de clasificación random Forest, para dispositivos móviles en plataforma Android.

Para estudios posteriores, se recomienda desarrollar el protocolo de recolección de datos con la aplicación de herramientas de cómputo como la NVIDIA Jetson Series o la Asus Tinker Board, que poseen características capaces de mejorar la cantidad, calidad y velocidad de la toma de datos.

Se recomienda que, para futuros trabajos con la aplicación de este algoritmo, se recolecten un número mínimo de 30 muestras por fallas, para garantizar la calidad del entrenamiento del algoritmo.

Se recomienda tener en cuenta los factores de temperatura y presión atmosférica, debido a que el motor debe estar a temperatura ideal de 97°C para alcanzar óptimas condiciones de funcionamiento.

Se recomienda la aplicación de herramientas de desarrollo en la nube para la mejor conexión entre los elementos del sistema.

Se recomienda que en conjunto con el desarrollo del algoritmo se construya el protocolo de comunicación entre elementos del sistema, para asegurar el funcionamiento óptimo del sistema.

4. Bibliografía

- Alaminos Fernández, Antonio F. «Árboles de decisión en R con Random Forest.» *Universidad de Alicante* (2023).
- Alonso Perez, Jose Manuel. *Técnicas del automóvil. Motores*. Madrid, 2009.
- Álvarez Flórez, Jesus Andres y Ismael Callejón Agramunt. *Motores alternativos de combustión interna*. Edicions de la UPC, S.L., 2005.
- Amazon. *Amazon*. s.f. 2023.
- Anonimo. 2005.
- Arciniega Zavala, Andres Sebastian y Joel Fabian Pastas Montes. «Desarrollo de una aplicación móvil para predecir las precipitaciones fluviales en la ciudad de Quito utilizando el algoritmo Random Forest.» (2023).
- Bosch. «Sistemas de Inyección Electrónica.» 2010.
- Cabezas Bejarano, Carlos y Edwin Damian Freire Poveda. «INVESTIGACIÓN DEL COMPORTAMIENTO MECÁNICO Y ELECTRÓNICO DE LOS INYECTORES MARCA BOSCH SERIE 0 445 110 290 Y 0 445 110 293 UTILIZADOS EN EL SISTEMA CRDI (COMMON RAIL DIESEL INJECTION).» (2018).
- Caldas Siavichay, Rolando Ismael. «Diseño de un algoritmo de visualización para el diagnóstico de fallas en motores de encendido provocado mediante la utilización de la transformada de Wavelet.» 2018. s.f.
- Caldas, Rolando. «Diseño de un algoritmo de visualización para el diagnóstico de fallas en motores de encendido provocado mediante la utilización de la transformada de Wavelet.» (2018).
- Calderon Orellana, Mario Fernando. «Identificación del fallo de los inyectores de un MEC mediante el análisis de las señales de los sensores CKP y CMP utilizando RNA.» 2022.
- Castro Mediavilla, Juan Jose y Marcelo Eduardo Gonzalez Torres. «Análisis comparativo del funcionamiento del inyector de un motor CRDI bajo diferentes alturas geograficas.» (2019).
- Contreras Urgiles, Wilmer Rafael y Jorge Esteban Fajardo Merchan. «Diseño y construcción de un sistema prototipo para determinar la cilindrada total de un motor ciclo Otto por un método no invasivo mediante Labview.» (2015).
- Contreras, Wilmer y Mauricio Arichavala. «Determinación de la presión máxima de compresión de un motor de encendido compresión de un motor de encendido compresión de un motor de encendido.» *INGENIUS* (2018).
- Data Science. *Random Forest: Bosque aleatorio. Definición y funcionamiento*. 2022.
- Delgado Calle, Esteban Homero. «Desarrollo de un algoritmo de diagnóstico para la detección de fallas mecánicas en motores de encendido provocado basados en la transformada Wavelet.» 2018.

Denso. *Denso UE*. s.f. 2023.

García Morales, J, y otros. «Multiple fault detection and isolation using artificial neural networks in sensors of an internal combustion engine.» *Engineering Applications of Artificial Intelligence* (2023).

Garzon Iñiguez, Galo Vinicio y Miguel Eduardo Urdiales Baculima. «Diseño e implementación de un sistema para determinar fallas mecánicas en motores de encendido provocado mediante redes neuronales artificiales.» Cuenca, 2017.

Gonzalez, Ligdi . *Aprendizaje Supervisado: Random Forest Classification*. 2018.

Hurtado Piña, Juan Jose, Juan Eduardo Morocho Cabrera y Juan Pablo Pinguil Lliguicota. «Desarrollo de un algoritmo para el diagnóstico del estado porcentual del rendimiento de los cilindros, mediante el análisis de la señal del sensor de posición del cigüeñal de un motor Nissan FD 6 cilindros.» (2014).

icon-icons. *icon-icons*. s.f. 2024.

Jimenez, Doris, Henry Paz - Arias y Andres Larco. «Desarrollo de un sistema inteligente para la clasificación de documentos ya digitalizados aplicando redes neuronales supervisadas.» (2015).

Komorska, Iwona, Zbigniew Wolczynski y Artur Borczuch. «Diagnosis of sensor faults in a combustion engine control system with the artificial neural network. .» *DIAGNOSTYKA* (2019).

Leon Japa, Rogelio Santiago y Jose Luis Maldonado Ortega. «Identificación de patrones de fallas mecánicas mediante redes neuronales artificiales para el diagnóstico de motores de encendido provocado.» Cuenca: Universidad Politecnica Salesiana, 2018. Proyecto tecnico.

Luna Avila, Juan Pablo y Jhon Andres Tapia Palacios. «Desarrollo de un prototipo computacional de diagnóstico de averías mediante recopilación de datos del sensor CKP de un motor Daewoo GM F16D3A.» (2021).

Manjarrez Montaña, Lino Francisco. «Relaciones neuronales para determinar la atenuacion del valor de la aceleracion maxima en superficie de sitios en roca para zonas de subduccion.» Ciudad de Mexico, 2014.

Matich, Damian Jorge. *Redes Neuronales: Conceptos Básicos y Aplicaciones*. 2001.

microscopio.pro. *microscopio.pro*. s.f. 2024.

Minitab. *Soporte Minitab*. s.f. 2023.

Moreno Navarrete, Marcos. «Identificación y diagnóstico de los MCIAMediante protocolos CAN-OBDII.» Ciudad de Mexico: UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO, 2016.

Raspberry Pi. *Raspberry Pi 4b*. s.f.

- Rovira de Antonio, Antonio y Marta Muñoz Domínguez. *Máquinas y motores térmicos: introducción a los motores alternativos y a las turbomáquinas térmicas*. Universidad Nacional de Educación a Distancia, 2016.
- Rovira de Antonio, Antonio y Marta Muñoz Dominguez. *Motores de combustion interna*. Madrid: UDED, 2015.
- Sanchez Camperos, Edgar Nelson y Alma Yolanda Alanis Garcia. *Redes Neuronales: Conceptos Fundamentales y Aplicaciones a Control Automatico*. Prentice Hall, 2006.
- solectroshop. *solectroshop*. s.f. 2023.
- Spectra Premium. *Spectra Premium*. s.f. 2024.
- Taglialatela, Fernando, y otros. «Determinación de los parámetros de combustión utilizando la velocidad del cigüeñal del motor.» *Mechanical Systems and Signal Processing* (2013).
- Torres Vargas, Enrique Adrian. «Sistema de diagnóstico automotriz mediante el análisis de emisiones contaminantes con la aplicación de redes neuronales para la detección de fallas.» Cuenca, 2017.
- Vélez de Guevara, Luis. *wordpress.com*. s.f. 2024.

5. ANEXOS

5.1. ANEXO 1

Graficas de residuos y comparaciones de parejas de tukey, para los atributos estadísticos más representativos.

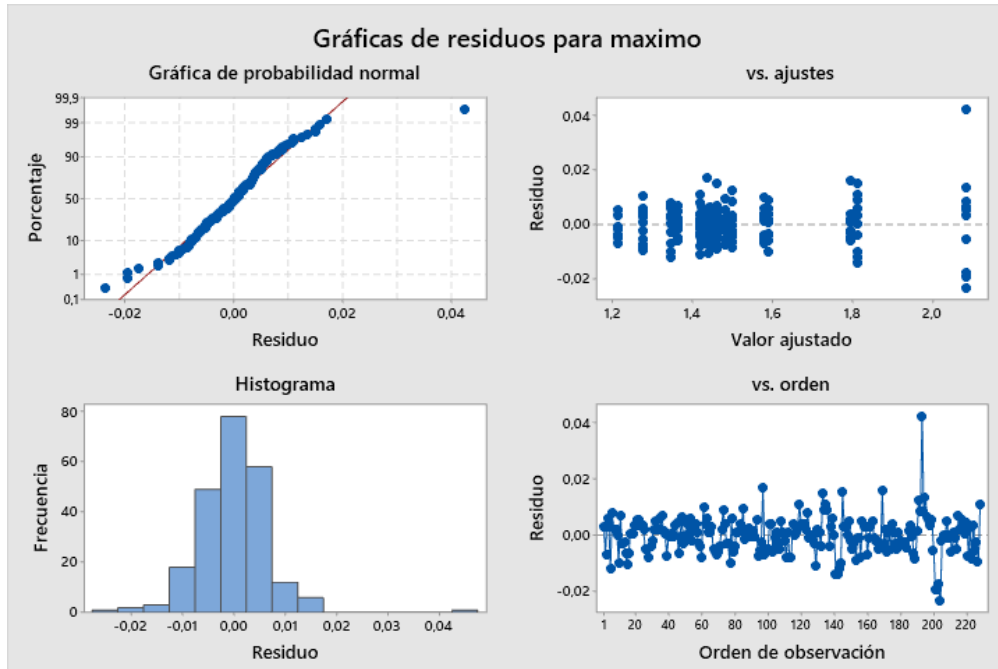


Figura 54 Grafica de residuos para atributo, Máximo.

Fuente: Autor

Comparaciones en parejas de Tukey

Agrupar información utilizando el método de Tukey y una confianza de 95%

resultado	N	Media	Agrupación
1600	12	2,08550	A
1100	12	1,81292	B
1400	12	1,79508	C
600	12	1,58908	D
500	12	1,57892	D
1500	12	1,50042	E
700	12	1,48250	F
1200	12	1,46175	G
200	12	1,45917	G
400	12	1,44475	H
100	12	1,43950	H
800	12	1,43592	H
300	12	1,42167	I
1000	12	1,41917	I
900	12	1,41792	I
1700	12	1,36392	J
0	12	1,34592	K
1800	12	1,27725	L
1300	12	1,21275	M

Las medias que no comparten una letra son significativamente diferentes.

Figura 55 Grafica de comparación de parejas de Tukey, para atributo máximo.

Fuente: Autor

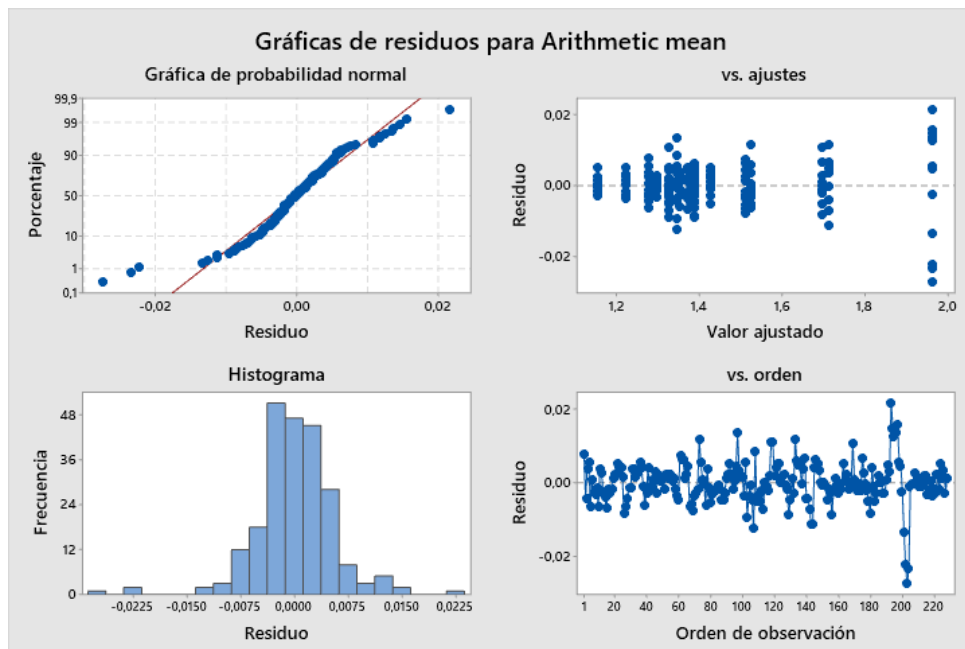


Figura 56 Gráfica de residuos para atributo, Media.

Fuente: Autor

Comparaciones en parejas de Tukey
Agrupar información utilizando el método de Tukey y una confianza de 95%

resultado	N	Media	Agrupación
1600	12	1,96333	A
1100	12	1,71317	B
1400	12	1,69617	C
600	12	1,52425	D
500	12	1,51267	E
1500	12	1,42800	F
1200	12	1,38875	G
700	12	1,38867	G
200	12	1,38742	G
400	12	1,37767	H
100	12	1,37275	H
300	12	1,35300	I
800	12	1,34450	I
900	12	1,32608	J
1000	12	1,32475	J
1700	12	1,29717	K
0	12	1,27833	L
1800	12	1,22367	M
1300	12	1,15367	N

Las medias que no comparten una letra son significativamente diferentes.

Figura 57 Gráfica de comparación de parejas de Tukey, para atributo media.

Fuente: Autor

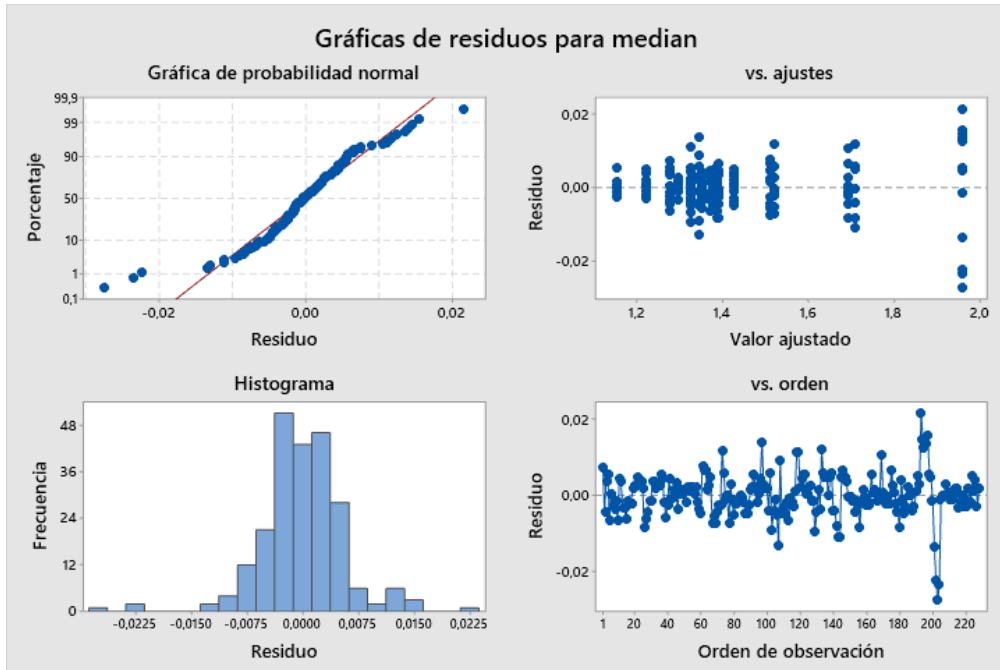


Figura 58 Grafica de residuos para atributo, Mediana.

Fuente: Autor

Comparaciones en parejas de Tukey

Agrupar información utilizando el método de Tukey y una confianza de 95%

resultado	N	Media	Agrupación
1600	12	1,95942	A
1100	12	1,71108	B
1400	12	1,69433	C
600	12	1,52217	D
500	12	1,51042	E
1500	12	1,42592	F
1200	12	1,38942	G
700	12	1,38867	G
200	12	1,38633	G
400	12	1,37675	H
100	12	1,37133	H
300	12	1,35167	I
800	12	1,34400	I
900	12	1,32575	J
1000	12	1,32450	J
1700	12	1,29617	K
0	12	1,27750	L
1800	12	1,22200	M
1300	12	1,15242	N

Las medias que no comparten una letra son significativamente diferentes.

Figura 59 Grafica de comparación de parejas de Tukey, para atributo mediana.

Fuente: Autor

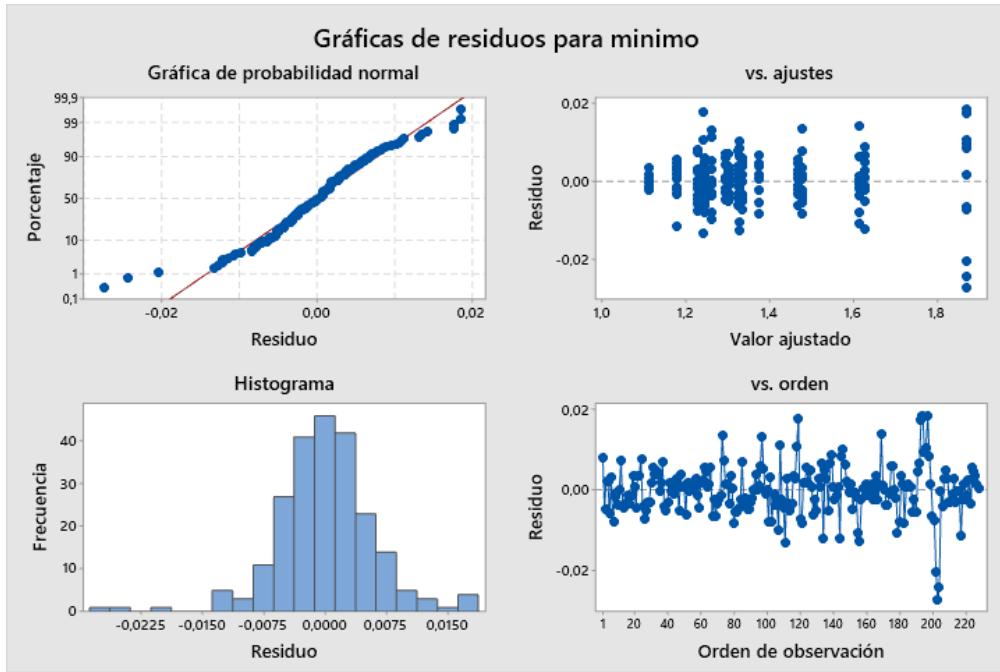


Figura 60 Grafica de residuos para atributo, mínimo.

Fuente: Autor

Comparaciones en parejas de Tukey

Agrupar información utilizando el método de Tukey y una confianza de 95%

resultado	N	Media	Agrupación
1600	12	1,87233	A
1100	12	1,62908	B
1400	12	1,61475	C
600	12	1,47933	D
500	12	1,46825	E
1500	12	1,37525	F
200	12	1,33508	G
1200	12	1,32658	G H
400	12	1,32417	H
100	12	1,32317	H
300	12	1,30000	I
700	12	1,29483	I
800	12	1,25975	J
1700	12	1,24600	K
1000	12	1,24308	K
900	12	1,24117	K
0	12	1,22667	L
1800	12	1,17933	M
1300	12	1,11233	N

Las medias que no comparten una letra son significativamente diferentes.

Figura 61 Grafica de comparación de parejas de Tukey, para atributo mínimo.

Fuente: Autor

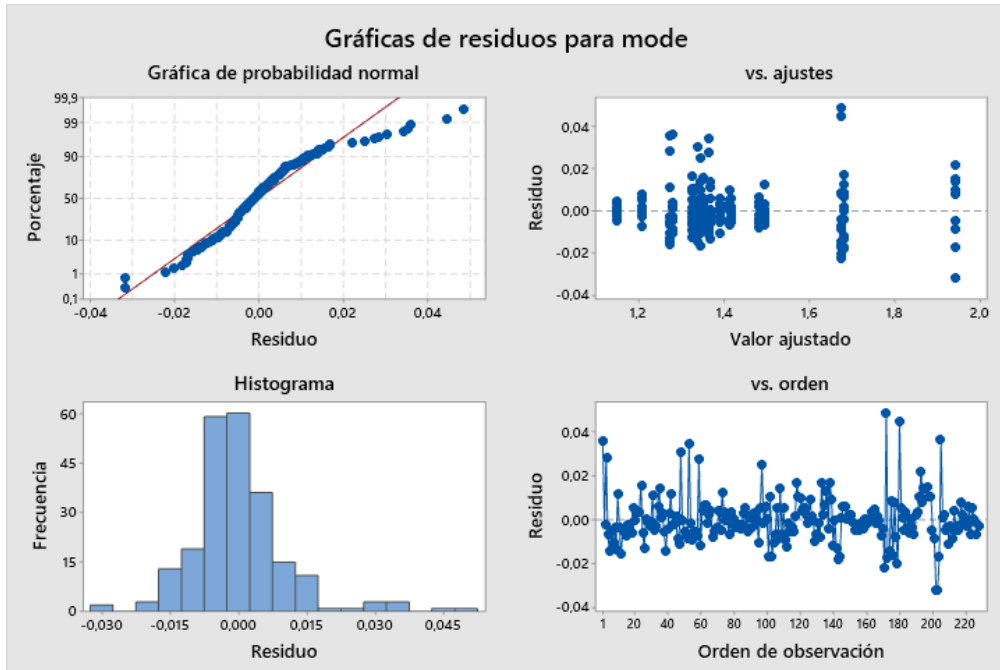


Figura 62 Grafica de residuos para atributo, moda.

Fuente: Autor

Comparaciones en parejas de Tukey

Agrupar información utilizando el método de Tukey y una confianza de 95%

resultado	N	Media	Agrupación
1600	12	1,94283	A
1100	12	1,68108	B
1400	12	1,67525	B
600	12	1,49342	C
500	12	1,48067	C
1200	12	1,41600	D
1500	12	1,41283	D
700	12	1,39033	E
200	12	1,36600	F
400	12	1,36350	F
100	12	1,35133	F G
800	12	1,34275	G H
300	12	1,33742	G H I
1000	12	1,32767	H I
900	12	1,32542	I
1700	12	1,27983	J
0	12	1,27250	J
1800	12	1,20592	K
1300	12	1,14642	L

Las medias que no comparten una letra son significativamente diferentes.

Figura 63 Grafica de comparación de parejas de Tukey, para atributo moda.

Fuente: Autor

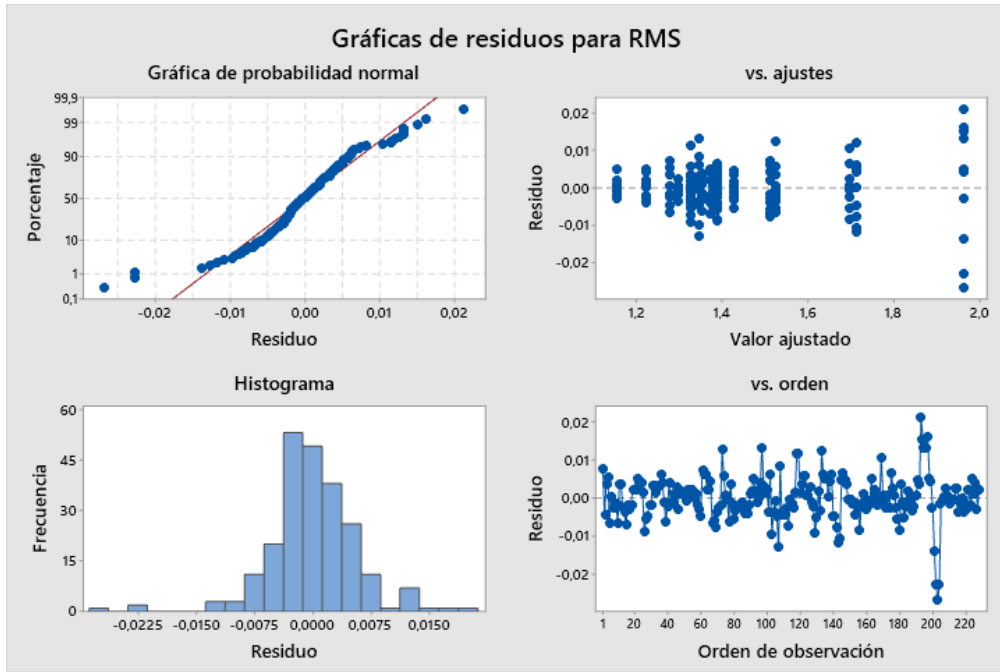


Figura 64 Grafica de residuos para atributo, RMS.

Fuente: Autor

Comparaciones en parejas de Tukey

Agrupar información utilizando el método de Tukey y una confianza de 95%

resultado	N	Media	Agrupación
1600	12	1,96375	A
1100	12	1,71375	B
1400	12	1,69642	C
600	12	1,52433	D
500	12	1,51275	E
1500	12	1,42825	F
1200	12	1,38942	G
700	12	1,38900	G
200	12	1,38767	G
400	12	1,37792	H
100	12	1,37283	H
300	12	1,35308	I
800	12	1,34467	I
900	12	1,32642	J
1000	12	1,32525	J
1700	12	1,29750	K
0	12	1,27850	L
1800	12	1,22383	M
1300	12	1,15383	N

Las medias que no comparten una letra son significativamente diferentes.

Figura 65 Grafica de comparación de parejas de Tukey, para atributo RMS.

Fuente: Autor

5.2.ANEXO 2

Este código en su conjunto está diseñado para adquirir datos analógicos, procesarlos, y enviar resultados a través de comunicación MQTT, como parte del sistema de diagnóstico.

```
import pandas as pd
from joblib import dump, load
import numpy as np
from scipy.stats import kurtosis, skew, mode
import time
import paho.mqtt.client as paho
from paho import mqtt
import threading
import time
import board
import busio
import adafruit_ads1x15.ads1115 as ADS
from adafruit_ads1x15.analog_in import AnalogIn
import os
```

Figura 66 Fragmento 1, código del proceso de diagnóstico.

Fuente: Autor

Se importan varias bibliotecas y módulos necesarios para el proyecto, incluyendo Pandas, joblib, numpy, scipy, paho MQTT (para la comunicación MQTT), y módulos relacionados con el hardware como board, busio, y adafruit_ads1x15.

```

python
data_enable = 0
data_in = 1

client = paho.Client(client_id="", userdata=None, protocol=paho.MQTTv5)
client.on_connect = on_connect

# enable TLS for secure connection
client.tls_set(tls_version=mqtt.client.ssl.PROTOCOL_TLS)
# set username and password
client.username_pw_set("modelo_ia", "modeloia2024")
# connect to HiveMQ Cloud on port 8883 (default for MQTT)
client.connect("72db3cb4ba8844ef812467a15ea848ce.s1.eu.hivemq.cloud", 8883)

# setting callbacks, use separate functions like above for better visibility
client.on_subscribe = on_subscribe
client.on_message = on_message
client.on_publish = on_publish

# subscribe to all topics of encyclopedia by using the wildcard "#"
client.subscribe("Rasp/Raspi/#", qos=1)

client.loop_start()

```

Figura 67 Fragmento 2, código del proceso de diagnóstico.

Fuente: Autor

Se configura un cliente MQTT para conectarse a un servidor MQTT en la nube (HiveMQ Cloud). Se establecen callbacks para diferentes eventos de MQTT, como conexión (on_connect), suscripción (on_subscribe), publicación (on_publish), y recepción de mensajes (on_message). Se suscribe a todos los temas bajo "Rasp/Raspi/#".

```

python
#modelo de clasificacion
model_filename = 'RandomForestClassifier.joblib'
# Cargar el modelo de clasificacion
classifier_loaded = load(model_filename)
print("Modelo cargado - Ahora lo puedes utilizar en tus predicciones")

# Crea el bus I2C
i2c = busio.I2C(board.SCL, board.SDA)

# Crea el objeto ADS1115
ads = ADS.ADS1115(i2c)

# Crea un canal analogico
chan = AnalogIn(ads, ADS.P0) # SELECCIONAR EL CANAL DE LECTURA

# Nombre del archivo donde se guardaran los datos
nombre_archivo = "T12F18.txt"

# Funcion para agregar datos al archivo
def guardar_datos(voltaje, timestamp):
    with open(nombre_archivo, "a") as archivo:
        archivo.write(f"{voltaje}V, {timestamp}\n")

while 1:

    if data_enable ==1:
        data_enable = 0
        data_entero = int.from_bytes(data_in, 'little')

    # Resto del código...

```

Figura 68 Fragmento 3, código del proceso de diagnóstico.

Fuente: Autor

La última parte del código define el modelo de clasificación, crea el bus I²C y el objeto ADS1115, selecciona el canal analógico y crea la función guardar_datos. El bucle principal verifica si data_enable está activado y, si lo está, lee el valor del sensor, calcula estadísticas y hace predicciones antes de enviar información a través de MQTT.

5.3.ANEXO 3

Este código muestra un flujo de trabajo típico para el entrenamiento, evaluación y aplicación de un modelo de clasificación utilizando datos reales y simulados

```

import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, classification_report
import matplotlib.pyplot as plt
from joblib import dump, load

print("Bibliotecas cargadas")

```

Figura 69 Fragmento 1, código de construcción y entrenamiento del algoritmo.

Fuente: Autor

Se importan las bibliotecas necesarias para el manejo de datos, creación de modelos de clasificación, evaluación de modelos y visualización. Además, se imprime un mensaje indicando que las bibliotecas han sido cargadas con éxito.

```

df = pd.read_excel('02_Tabla_Grande.xlsx')
X = df.drop(['total samples', 'resultado'], axis=1)
y = df['resultado']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=
classifier = RandomForestClassifier(random_state=42)
classifier.fit(X_train, y_train)

model_filename = 'RandomForestClassifier.joblib'
dump(classifier, model_filename)
print(f"Modelo de Clasificación guardado como '{model_filename}'")

```

Figura 70 Fragmento 2, código de construcción y entrenamiento del algoritmo.

Fuente: Autor

Se leen los datos desde un archivo Excel y se preparan para el entrenamiento de un modelo de clasificación. Se divide el conjunto de datos en conjuntos de entrenamiento y prueba, se crea un modelo de clasificación (RandomForestClassifier) y se entrena. Luego, se guarda el modelo en un archivo llamado 'RandomForestClassifier.joblib'.

```

classifier_loaded = load(model_filename)
print("Modelo cargado - Ahora lo puedes utilizar en tus predicciones")

y_pred_class = classifier_loaded.predict(X_test)
accuracy = accuracy_score(y_test, y_pred_class)
print("Precisión del modelo de clasificación:", accuracy)
print("Reporte de clasificación:\n", classification_report(y_test, y_pred_class))

```

Figura 71 Fragmento 3, código de construcción y entrenamiento del algoritmo.

Fuente: Autor

Se carga el modelo previamente guardado ('RandomForestClassifier.joblib') y se evalúa su precisión utilizando el conjunto de prueba. Se imprime la precisión del modelo y un informe detallado de la clasificación.

```
import numpy as np
from scipy.stats import kurtosis, skew, mode

limite_inferior = 1.27
limite_superior = 1.28

voltage_data = []
for _ in range(872): # Cambia 1 por el número de valores que quieres generar
    valor_aleatorio = np.random.uniform(limite_inferior, limite_superior)
    voltage_data.append(valor_aleatorio)

print("Longitud de voltage_data:", len(voltage_data))

arithmetic_mean = np.mean(voltage_data)
root_mean_square = np.sqrt(np.mean(np.square(voltage_data)))
# ... (cálculos estadísticos adicionales)

new_data = pd.DataFrame([
    "Arithmetic mean": arithmetic_mean,
    "RMS": root_mean_square,
    # ... (otras columnas)
]), columns=['Arithmetic mean', 'RMS', 'Standard dev', 'Variance', 'kurtosis', 'media',
            'sumatoria', 'skewness', 'maximo', 'minimo', 'range'])
```

Figura 72 Fragmento 4, código de construcción y entrenamiento del algoritmo.

Fuente: Autor

Se generan datos simulados (voltajes) y se realiza un análisis estadístico sobre estos datos. Se utilizan funciones de numpy y scipy.stats para calcular la media, la raíz cuadrada de la media cuadrática, la desviación estándar, la varianza, la kurtosis, la mediana, la moda, la suma, la asimetría, el máximo, el mínimo, el rango y la cantidad total de muestras.

```
predicted_result = classifier_loaded.predict(new_data)
print("Predicción del resultado:", predicted_result[0])
```

Figura 73 Fragmento 5, código de construcción y entrenamiento del algoritmo.

Fuente: Autor

Se realiza una predicción utilizando el modelo de clasificación cargado ('RandomForestClassifier.joblib') sobre los datos simulados. La predicción resultante se imprime en la consola.

5.4.ANEXO 4

Este código es una implementación de una aplicación Android en Java que utiliza la biblioteca HiveMQ para conectarse a un servidor MQTT (Message Queuing Telemetry Transport) y realizar operaciones de conexión, publicación y suscripción.

```
import static com.hivemq.client.mqtt.MqttGlobalPublishFilter.ALL;
import static java.nio.charset.StandardCharsets.UTF_8;
import androidx.annotation.RequiresApi;
import androidx.appcompat.app.AppCompatActivity;
import android.os.Build;
import android.os.Bundle;
import android.os.Handler;
import android.os.Looper;
import android.util.Log;
import android.view.View;
import android.widget.Button;
import android.widget.ProgressBar;
import android.widget.TextView;
import android.widget.Toast;
import com.hivemq.client.mqtt.mqtt5.Mqtt5AsyncClient;
import com.hivemq.client.mqtt.mqtt5.Mqtt5Client;
import com.hivemq.client.mqtt.mqtt5.message.connect.connack.Mqtt5ConnAck;
import java.util.UUID;
```

Figura 74 Fragmento 1, código de construcción del aplicativo.

Fuente: Autor

Se importan las bibliotecas necesarias para la aplicación, incluida la biblioteca de HiveMQ para MQTT.

```
public class MainActivity extends AppCompatActivity {
    // Declaración de variables y objetos de la aplicación, incluyendo un cliente MQTT
}
```

Figura 75 Fragmento 2, código de construcción del aplicativo.

Fuente: Autor

Se declaran variables y objetos

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    // Configuración de la interfaz de usuario y asignación de eventos a los botones.
}

```

Figura 76 Fragmento 3, código de construcción del aplicativo.

Fuente: Autor

Se realiza la inicialización de la interfaz de usuario y se asignan los eventos a los botones.

```

Mqtt5Client client = Mqtt5Client.builder()
    .identifier(UUID.randomUUID().toString())
    .serverHost("72db3cb4ba8844ef812467a15ea848ce.s1.eu.hivemq.cloud")
    .serverPort(8883)
    .sslWithDefaultConfig()
    .simpleAuth()
    .username("modelo_ia")
    .password("modeloia2024".getBytes())
    .applySimpleAuth()
    .build();

```

Figura 77 Fragmento 4, código de construcción del aplicativo.

Fuente: Autor

Aquí se configura el cliente MQTT utilizando HiveMQ. Se especifica el identificador del cliente, el servidor al que se conectará, el puerto, la configuración SSL y las credenciales de autenticación.

```

Handler HiveMQ_DataIn;
final int handlerState = 0;
private StringBuilder recDataString = new StringBuilder();
final String StrHiveMQ_TopicSub = "Rasp/Android/#";
String StrHiveMQ_TopicSubIn = "some payload";
String StrHiveMQ_TopicPub = "some payload";
String str_payload = "some payload";
String StrDataHiveMQ_Out = "some payload";

```

Figura 78 Fragmento 5, código de construcción del aplicativo.

Fuente: Autor

Se declara un manejador (Handler) y algunas variables relacionadas con la interfaz de usuario y la comunicación con el servidor MQTT.

```

public String ConectarHiveMQ() {
    // Método para conectar al servidor HiveMQ
}

public void PublicarHiveMQ() {
    // Método para publicar mensajes en HiveMQ
}

@RequiresApi(api = Build.VERSION_CODES.N)
public void SuscribirseHiveMQ() {
    // Método para suscribirse a un tema en HiveMQ
}

```

Figura 79 Fragmento 6, código de construcción del aplicativo.

Fuente: Autor

Estos métodos realizan las operaciones de conexión, publicación y suscripción utilizando el cliente MQTT de HiveMQ.

```

IdBtnConect.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Manejador del evento de clic en el botón de conexión
    }
});

IdBtnSuscrip.setOnClickListener(new View.OnClickListener() {
    @RequiresApi(api = Build.VERSION_CODES.N)
    @Override
    public void onClick(View v) {
        // Manejador del evento de clic en el botón de suscripción
    }
});

IdBtn_GetData.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        // Manejador del evento de clic en el botón para obtener datos
    }
});

```

Figura 80 Fragmento 7, código de construcción del aplicativo.

Fuente: Autor

Estos eventos de clic en los botones ejecutan diferentes acciones según el botón presionado.

