



ESPE
UNIVERSIDAD DE LAS FUERZAS ARMADAS
INNOVACIÓN PARA LA EXCELENCIA

**Sistema de mantenimiento predictivo para máquinas eléctricas en procesos industriales en el
escenario de la industria 4.0**

Pillajo Ñauñay, Martha Cecilia y Vivanco Correa, Jessica Andrea

Departamento de Eléctrica y Electrónica

Centro de Posgrados

Maestría en Electrónica y Automatización Mención Redes Industriales

Trabajo de titulación previo a la obtención del título de Magister en Electrónica y Automatización,
mención Redes Industriales.

Ing. Rodríguez Rivera, José María Mgs.

12 de agosto del 2022

Latacunga

Reporte de verificación de contenido



TESIS PILLAJO_VIVANCO_V2.pdf

Scanned on: 14:8 August 11, 2022 UTC



Identical Words	930
Words with Minor Changes	0
Paraphrased Words	0
Omitted Words	0

Ing. José María Rodríguez Mgs.

Director



Vicerrectorado de Investigación, Innovación y Transferencia de Tecnología

Centro de Posgrados

Certificación

Certifico que el trabajo de titulación: **“Sistema de Mantenimiento Predictivo para Máquinas Eléctricas en Procesos Industriales en el Escenario de la Industria 4.0”** fue realizado por las señoritas **Pillajo Ñauñay, Martha Cecilia, Vivanco Correa, Jessica Andrea**, el mismo que cumple con los requisitos legales, teóricos, científicos, técnicos y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, además fue revisado y analizado en su totalidad, por la herramienta de prevención y/o verificación de similitud de contenidos; razón por la cual me permito acreditar y autorizar para que lo sustenten públicamente.

Latacunga, 12 de agosto de 2022

Firma:

Ing. Rodríguez Rivera, José María Mgs.

Director

C.C.: 0501870398



Vicerrectorado de Investigación, Innovación y Transferencia de Tecnología

Centro de Posgrados

Responsabilidad de autoría

Nosotras, **Pillajo Ñauñay, Martha Cecilia, Vivanco Correa, Jessica Andrea**, con cédulas de ciudadanía n° 0604243998, n°1719383943, declaramos que el contenido, ideas y criterios del trabajo de titulación: **“Sistema de Mantenimiento Predictivo para Máquinas Eléctricas en Procesos Industriales en el Escenario de la Industria 4.0”**, es de nuestra autoría y responsabilidad, cumpliendo con los requisitos legales, teóricos, científicos, técnicos, y metodológicos establecidos por la Universidad de las Fuerzas Armadas ESPE, respetando los derechos intelectuales de terceros y referenciando las citas bibliográficas.

Latacunga, 03 de agosto de 2022

Firma:

Pillajo Ñauñay, Martha Cecilia

C.C.: 0604243998

Vivanco Correa, Jessica Andrea

C.C.: 1719383943



Vicerrectorado de Investigación, Innovación y Transferencia de Tecnología

Centro de Posgrados

Autorización de publicación

Nosotras: **Pillajo Ñauñay, Martha Cecilia, Vivanco Correa, Jessica Andrea**, con cédulas de ciudadanía n° 0604243998, n° 1719383943, autorizo a la Universidad de las Fuerzas Armadas ESPE publicar el trabajo de titulación **“Sistema de Mantenimiento Predictivo para Máquinas Eléctricas en Procesos Industriales en el Escenario de la Industria 4.0”**, en el Repositorio Institucional, cuyo contenido, ideas y criterios son de nuestra responsabilidad.

Latacunga, 03 de agosto de 2022

Firma:

Pillajo Ñauñay, Martha Cecilia

C.C.: 0604243998

Vivanco Correa, Jessica Andrea

C.C.: 1719383943

Dedicatoria

Este trabajo está dedicado a mi padre, quien me enseñó que la constancia es la única manera para alcanzar las metas propuestas. También está dedicado a mi madre, quien me inspira a plantearme siempre nuevos objetivos y a alcanzarlos con su apoyo.

Pillajo Ñauñay, Martha Cecilia

Principalmente a Dios quien ha sido mi guía, quien me ha dado la fuerza para continuar en este proceso de cumplir uno de mis anhelos más deseados. A mis Queridos Padres quienes con su amor, paciencia y esfuerzo me han permitido llegar a cumplir uno más de mis sueños, gracias por enseñarme lo que es el esfuerzo y la valentía para no detenerme ante las adversidades y seguir siempre adelante.

Vivanco Correa, Jessica Andrea

Agradecimiento

Quiero agradecer en estas líneas la ayuda que muchas personas y colegas me han brindado durante el proceso de investigación y redacción de este trabajo. En primer lugar, quisiera agradecer a mis padres que me han ayudado y apoyado durante este trayecto, a mi tutor, José María Rodríguez, por haberme orientado en los momentos que necesité sus consejos.

También quiero agradecer a la Universidad de las Fuerzas Armadas-ESPE, directivos y profesores por la organización del programa de Maestría en Electrónica y Automatización.

Pillajo Ñauñay, Martha Cecilia

Quiero dar gracias Dios, por bendecirme cada día, por darme salud y vida para poder seguir cultivándome y así poder cumplir todos mis sueños.

Gracias a mis padres y a mis hermanos, por confiar en mí, por apoyarme de forma incondicional, por ayudarme a seguir adelante, gracias por aconsejarme y por inculcarme valores y principios para que cada día siga siendo una mejor persona.

Finalmente quiero agradecer a los docentes de la Unidad de las Fuerzas Armadas ESPE, por haber compartido sus conocimientos conmigo a lo largo de toda la preparación de esta profesión, gracias por su paciencia, por su dedicación, por su apoyo y por su amistad.

Vivanco Correa, Jessica Andrea

ÍNDICE DE CONTENIDO

Carátula.....	1
Reporte de verificación de contenido	2
Certificación	3
Responsabilidad de autoría	4
Autorización de publicación	5
Dedicatoria.....	6
Agradecimiento	7
Índice de contenido	8
Índice de figuras.....	12
Índice de tablas.....	14
Resumen	15
Abstract.....	16
Capítulo I: Contenidos generales	17
Antecedentes investigativos.....	17
Planteamiento del problema	18
Justificación, importancia y alcance del proyecto	19
Objetivos del proyecto	20
<i>Objetivo general</i>	20
<i>Objetivos específicos</i>	20
Hipótesis de investigación	20
Categorización de las variables de investigación.....	20
<i>Variable Independiente</i>	20
<i>Variable Dependiente</i>	20

Capítulo II: Fundamento teórico	22
Introducción	22
Industria 4.0.....	22
<i>Campos de aplicación.....</i>	25
Internet de las cosas (IoT).....	25
<i>IOT como la red de redes</i>	26
<i>Elementos del IOT/.....</i>	26
<i>Objetivo del IoT.....</i>	27
Cloud computing.....	27
<i>Características del cloud computing.....</i>	28
<i>Modelos de servicio del cloud computing</i>	29
IaaS (infraestructure as a service).....	29
PaaS (plataform as a service).....	29
SaaS (Software as a Service).....	29
<i>Beneficios del cloud computing.....</i>	29
<i>Riesgos en entornos cloud computing</i>	30
<i>Tipos de cloud computing</i>	31
Nubes Públicas.	31
Nubes Privadas.....	31
Nubes Híbridas	32
Arquitectura de la nube.....	32
Protocolo MQTT.....	32
Arquitectura del sistema	34
Componentes hardware del sistema	34

Broker MQTT	34
Mosca	34
Mosquitto	35
NodeRed	35
Edición de flujo basado en un navegador	35
Node.js	35
Matriz de confusión	35
Falsos negativos (FN)	36
Falsos positivos (FP)	37
Verdaderos positivos (VP)	37
Verdaderos Negativos (VN)	37
Accuracy (Exactitud)	37
Sensibilidad	37
Especificidad	37
Capítulo III: Metodología	38
Descripción del proceso	38
Método de desarrollo de un sistema de mantenimiento predictivo	39
Adquisición de datos	40
Comunicación entre analizador de energía y módulo Logo 8	40
Comunicación entre módulo Logo 8 y PLC	41
Comunicación entre Variador y PLC	41
Comunicación entre PLC y raspberry	41
Comunicación entre raspberry y servidor web	41
Almacenamiento de datos en MySQL	41

Desarrollo del sistema	41
<i>Preprocesamiento de datos</i>	42
<i>Identificación de indicadores de condición</i>	46
<i>Entrenamiento del modelo</i>	49
<i>Implementación e integración</i>	54
Capítulo IV: Resultados	55
Interfaz de sistema de supervisión	55
Sistema en funcionamiento normal	56
Sistema con código de falla 1	56
Sistema con código de falla 2	57
Resumen de reportes de mantenimiento	58
<i>Reportes de mantenimiento antes de la implementación del sistema</i>	59
<i>Reportes de mantenimiento con el sistema implementado</i>	59
Capítulo V: Conclusiones y recomendaciones	60
Conclusiones	60
Recomendaciones	62
Bibliografía	63
Anexos	67

ÍNDICE DE FIGURAS

Figura 1 <i>Industria Conectada 4.0</i>	23
Figura 2 <i>Funcionamiento del protocolo MQTT</i>	33
Figura 3 <i>Distribución matriz de confusión</i>	36
Figura 4 <i>Sistema de ósmosis inversa</i>	39
Figura 5 <i>Método de desarrollo de un sistema de mantenimiento predictivo</i>	39
Figura 6 <i>Arquitectura de red</i>	40
Figura 7 <i>Ingreso de datos en Matlab</i>	42
Figura 8 <i>Asistente de importación de datos</i>	43
Figura 9 <i>Interfaz de la herramienta Feature Designer en Matlab</i>	43
Figura 10 <i>Representación gráfica de los datos puros</i>	44
Figura 11 <i>Aplicación de filtro en el conjunto de datos</i>	45
Figura 12 <i>Representación gráfica de los datos filtrados</i>	45
Figura 13 <i>Procesamiento de datos en función del tiempo</i>	46
Figura 14 <i>Menú de configuración en el dominio del tiempo</i>	47
Figura 15 <i>Ventana de configuración de modelo</i>	47
Figura 16 <i>Modelo auto regresivo de espectro en función de la frecuencia</i>	48
Figura 17 <i>Configuración de modelo auto regresivo en un rango de frecuencia</i>	49
Figura 18 <i>Ventana de selección de características</i>	50
Figura 19 <i>Ventana de confirmación de datos seleccionados</i>	51
Figura 20 <i>Ventana de la herramienta Classification Learner for Predictive Maintenance</i>	52
Figura 21 <i>Resultados de aprendizaje del modelo</i>	53
Figura 22 <i>Modelo obtenido en Matlab</i>	54
Figura 23 <i>Placa de motor Baldor</i>	55

Figura 24 <i>Interfaz del sistema en funcionamiento normal</i>	56
Figura 25 <i>Interfaz del sistema con codificación de fallo 1</i>	57
Figura 26 <i>Interfaz del sistema con codificación de fallo 2</i>	58

ÍNDICE DE TABLAS

Tabla 1 <i>Operacionalización de las variables</i>	21
Tabla 2 <i>Resumen de mantenimientos previa implementación del sistema</i>	59
Tabla 3 <i>Resumen de mantenimientos posterior a la implementación del sistema</i>	59

Resumen

En el presente trabajo de titulación se busca clasificar las fallas más comunes presentadas por una bomba centrífuga dentro de un sistema de ósmosis inversa, mediante el desarrollo de un modelo de mantenimiento predictivo basado en las características de operación del equipo. El objetivo es detectar anomalías y patrones de fallos en el equipo para anticiparse a posibles errores que se puedan producir, con el fin de maximizar los tiempos de operación, al minimizar el mantenimiento correctivo no planificado, esto permite disminuir al mínimo las fallas no previstas sin incrementar la cantidad de inspecciones rutinarias. El desarrollo del sistema contempla cinco etapas: adquisición de datos, procesamiento, identificación de indicadores de condición, entrenamiento del modelo y finalmente implementación e integración. En la primera etapa se implementan protocolos de comunicación industrial como: Modbus TCP, Profinet, S7 y el protocolo MQTT para conectividad IoT, durante la etapa de procesamiento se determinó que el parámetro que presenta mayor discriminación para la detección de fallas es la señal de corriente, la segunda, tercera y cuarta etapa se desarrolla utilizando las herramientas *DiagnosticFeatureDesigner* y *Classification Learner*, dedicadas al desarrollo de algoritmos de mantenimiento predictivo dentro de *Matlab*. La última etapa corresponde a la implementación e integración del modelo, en donde se presenta una interfaz gráfica de usuario basada en una plataforma web para visualizar en tiempo real los resultados del sistema. Se recomienda para futuros trabajos de investigación añadir dispositivos que permitan obtener datos de vibración y de espectro térmico, parámetros utilizados para definir la vida útil restante (RUL) del equipo.

Palabras Clave: Mantenimiento predictivo, IoT, MQTT, *Cloud Computing*, Industria 4.0.

Abstract

This degree work seeks to classify the most common failures presented by a centrifugal pump within a filtration system, through the development of a predictive maintenance model based on the operating characteristics of the equipment. The objective is to detect anomalies and failure patterns in the equipment to anticipate possible errors that may occur, in order to maximize operating times by minimizing unplanned corrective maintenance, this allows minimizing unplanned failures without increasing the number of routine inspections. The development of the system contemplates five stages: data acquisition, processing, condition indicator identification, model training and finally implementation and integration. In the first stage, industrial communication protocols such as: Modbus TCP, Profinet, S7 and the MQTT protocol for IoT connectivity are implemented; during the processing stage it was determined that the parameter that presents the greatest discrimination for fault detection is the current signal; the second, third and fourth stages are developed using the Diagnostic Feature Designer and Classification Learner tools, dedicated to the development of predictive maintenance algorithms within MATLAB. The last stage corresponds to the implementation and integration of the model, where a graphical user interface based on a web platform is presented to visualize in real time the results of the system. It is recommended for future research work to add devices to obtain vibration and thermal spectrum data, parameters used to define the remaining useful life (RUL) of the equipment.

Keywords: Predictive maintenance, IoT, MQTT, Cloud Computing, Industry 4.0.

Capítulo I

Contenidos generales

Antecedentes investigativos

A finales del siglo XVIII y comienzo del XIX y coincidiendo con la revolución industrial provocada principalmente por la aparición de la máquina de vapor, la introducción de las primeras máquinas en las industrias, así como la industrialización de los procesos de extracción del hierro, surge el concepto de “fallo” y de “mantenimiento” con la necesidad de realizar la reparación de las máquinas tras la aparición de los fallos para aumentar la competitividad de las plantas industriales (Martínez F., 2015).

El mantenimiento predictivo surge como complemento del correctivo y del preventivo. En este tipo de mantenimiento una serie de parámetros son monitoreados y analizados para determinar posibles anomalías. El proceso esencialmente se centra en generar estimados sobre el estado de un componente en particular. Cuando se tienen muchas cantidades de variables es posible generar un modelo matemático que represente de la forma más precisa posible a la realidad. Otros tipos de procesos requieren pruebas experimentales como la clasificación por algoritmos o el uso de redes neuronales. Estos modelos trabajan con los datos extraídos de los datos históricos de las máquinas (De los Ríos G. 2019).

En el sector industrial existen varios tipos de mantenimiento, uno de ellos es el mantenimiento predictivo que se basa en la evaluación de condiciones y del estado de una máquina. Y en base a esto recomienda si se debe intervenir o no, el objetivo principal de este tipo de mantenimiento es garantizar disponibilidad de maquinaria y equipo crítico ahorrando recursos. La monitorización de máquinas no es nueva, anteriormente los datos solo se almacenaban, pero no se usaban y estaban limitados a estados de las máquinas. (Candanedo, González, & Muñoz, 2018).

El concepto de Industria 4.0 la obtención y análisis de datos es el pilar fundamental, si se aplican técnicas de modelado y procesamiento adecuado, todos los datos obtenidos se transformarán en

información de utilidad para toma de decisiones. Según Gilchrist (2016) la Industria 4.0 se caracteriza por centrarse en mejorar modelos de negocios y eficiencia operacional.

Una de las técnicas de mantenimiento predictivo se basa en el análisis de variables para identificar y estimar fallos potenciales en equipos. Uno de los problemas que se presenta en el análisis de estas variables es la forma de usar estos datos, uno de los modelos que mejor resultados ha dado son lo que priorizan la obtención de patrones dentro de un ciclo. (Candanedo, González, & Muñoz, 2018).

Planteamiento del problema

Las industrias siempre han tenido que tolerar periodos de inactividad no planificada debido a fallas intempestivas, cuando esto sucede las empresas tienen que abordar el problema tan rápido como sea posible y luego poner en marcha de nuevo los equipos, teniendo que soportar las consecuencias derivadas del daño inicial.

Ante esta situación las empresas tratan de cumplir con todos los requisitos de mantenimiento del equipo con el fin de minimizar tales casos, sin embargo, con el gran número de equipos a monitorear esta tarea se complica, con numerosos equipos en planta, el tiempo de inactividad hoy en día, tiene un impacto en la productividad, el rendimiento, la rentabilidad y el desgaste de los equipos.

El sistema a desarrollar, se enfoca en monitorear y evaluar un proceso de sistema de ósmosis inversa que es parte de una planta de tratamiento de agua implementada en una fábrica de procesamiento de caucho, el sistema de ósmosis inversa es la etapa final de un proceso secuencial de filtración, ésta implica el uso de membranas para su funcionamiento, las cuales deben ser cambiadas cada seis meses o cuando los parámetros del agua tratada no cumplen requerimientos mínimos, la planta funciona las 24 horas, por lo que, las actividades de mantenimiento correctivo en cualquiera de las etapas implican un tiempo de parada en la planta a nivel general, en la etapa de la ósmosis los tiempos de detección de fallas tienen un rango de entre 20 a 30 minutos previo a su corrección, el

presente sistema busca disminuir ese tiempo, tipificando las fallas de proceso y eléctricas más comunes presentadas durante su funcionamiento.

Justificación, importancia y alcance del proyecto

Los avances tecnológicos y el desarrollo progresivo de lo que respecta a la Industria 4.0, ha impulsado el crecimiento de análisis de datos y en la inteligencia artificial. Sería de mucha ayuda analizar las variables que provienen de los componentes de un proceso automatizado, para desarrollar un sistema de mantenimiento predictivo que esté encaminado en el análisis de las variables generadas, por ejemplo, el análisis de componentes fundamentales de un motor eléctrico. A diferencia de los sistemas tradicionales de mantenimiento predictivo, el análisis de variables en este tipo de mantenimiento que se propone usa datos que ya existen en un proceso automatizado. También resulta mucho más favorable de manera económica y práctica, ya que requiere menos recursos económicos y se realiza de forma automática. La Industria 4.0 obligará a integrar elementos y tecnologías de vanguardia dentro del mundo industrial como Internet de las Cosas (IoT), *Cloud Computing*, *Business Inteligencia*, Realidad Virtual/Aumentada o Ciberseguridad, entre otras. La implementación de todo ello supone un reto muy importante para las empresas. (Candanedo, González, & Muñoz, 2018).

El presente proyecto de investigación está limitado al desarrollo de un sistema de mantenimiento predictivo para obtener alertas de posibles fallas a producirse en las máquinas eléctricas de la planta de tratamiento de aguas residuales en la fábrica Indecaúcho ubicada en Cayambe, mediante el análisis de las variables obtenidas a través de los sensores presentes en el proceso. El sistema generará notificaciones para que acciones correctivas sean ejecutadas por el personal a cargo.

Objetivos del proyecto

Objetivo general

- Diseñar un sistema de mantenimiento predictivo para máquinas eléctricas en procesos industriales en el escenario de la Industria 4.0.

Objetivos específicos

- Definir las variables de proceso a evaluar para el análisis del comportamiento de la máquina.
- Implementar los protocolos de comunicación para integrar todos los dispositivos del sistema.
- Desarrollar el modelo matemático para la detección de valores anormales.
- Analizar los valores atípicos encontrados para detectar la posible falla.
- Diseñar una interfaz gráfica de usuario para visualizar los resultados obtenidos.

Hipótesis de investigación

La implementación del sistema de mantenimiento predictivo para máquinas eléctricas permitirá reducir la ocurrencia de fallas intempestivas en procesos industriales en el escenario de la Industria 4.0.

Categorización de las variables de investigación

A partir, de la hipótesis planteada se logra identificar dos variables:

Variable independiente

Sistema de mantenimiento predictivo para máquinas eléctricas.

Variable dependiente

Ocurrencia de fallas intempestivas en procesos industriales en el escenario de la Industria 4.0.

Tabla 1*Operacionalización de las variables*

VARIABLES	TIPO	DEFINICIÓN CONCEPTUAL	DEFINICIÓN OPERACIONAL	DIMENSIONES	INDICADORES
Sistema de mantenimiento predictivo para máquinas eléctricas	Independiente	Conjunto de procedimientos establecidos para anticipar la ocurrencia de fallas en máquinas eléctricas.	Los modelos matemáticos utilizados para la detección de valores anormales con el fin de detectar fallas eléctricas en maquinaria.	<p>Confiabilidad de detección de fallas.</p> <p>Tiempo de respuesta.</p> <p>Costo de implementación.</p>	- Porcentaje de aciertos del sistema para la clasificación de fallas específicas
Ocurrencia de fallas intempestivas en procesos industriales en el escenario de la industria 4.0.	Dependiente	Variables del sistema que presenten un comportamiento anormal.	Parámetros eléctricos o de proceso que presenten valores atípicos en su comportamiento normal y que deben ser reconocidos por el sistema implementado.	<p>Parámetros eléctricos con valores atípicos.</p> <p>Parámetros de proceso con valores atípicos.</p> <p>Cantidad de fallas</p>	- Fallas específicas detectadas

Capítulo II

Fundamento teórico

Introducción

Durante la primera revolución industrial, con el advenimiento de la máquina de vapor, fue un hecho fundamental que cambió la producción industrial en ese momento a través de la introducción de maquinaria que aumentaba la producción y reducía el tiempo de funcionamiento asociado a la actividad humana.

Luego se desarrolló la segunda revolución industrial que nació a partir del descubrimiento de nuevas fuentes de energía como el gas, el petróleo y la electricidad.

Luego, la Tercera Revolución Industrial introdujo el uso de la electrónica y la tecnología de la información para mejorar la producción automatizada.

Finalmente, la cuarta revolución industrial está teniendo lugar hoy. Al igual que las tres revoluciones anteriores, esta nueva revolución se basa en la aplicación de nuevas tecnologías a los procesos industriales, especialmente el uso de sistemas cibernéticos.

Industria 4.0

La Industria 4.0 se basa en el llamado "sistema de producción cibernético". Entre ellos, el proceso de fabricación, ya sea físico o biológico, está controlado o supervisado por algoritmos estrechamente integrados con Internet. Para ello, se apoyan en un modelo digital del proceso productivo y el intercambio de datos durante la producción, entre productos y máquinas y/o entre los diferentes actores de la cadena productiva. (Bidet M., 2016).

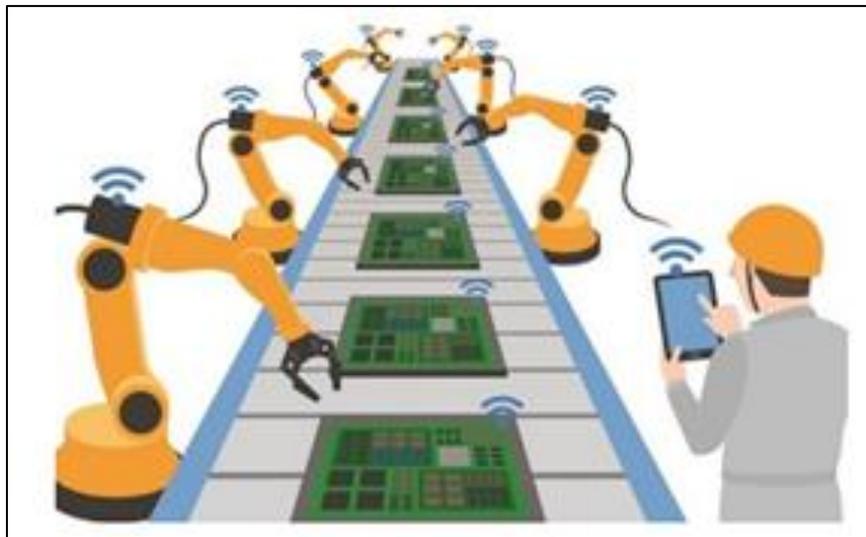
Actualmente vivimos la llamada Cuarta Revolución Industrial, o Industria 4.0, término acuñado por el gobierno alemán en 2016 para resaltar la necesidad de automatizar las cadenas industriales mediante sensores, redes inalámbricas y comunicaciones 5G. Los avances en nuevos microprocesadores

de bajo costo, la cantidad de sensores en el mercado actual, la facilidad de procesamiento y almacenamiento de datos, así como el análisis de datos a través de *big data* son una base fundamental para las tecnologías que están cambiando la industria hoy en día. (Sang et al., 2020).

El objetivo principal es transformar la maquinaria o los activos industriales existentes para que registren todos los parámetros y puedan auto diagnosticarse para mejorar el rendimiento y mejorar el mantenimiento. Así, con una plataforma industrial, es posible monitorear todos los procesos y registrar el estado de las máquinas y los parámetros del proceso productivo en tiempo real. Sin embargo, para llegar al punto en el que se puedan ver los beneficios de la Industria 4.0, las empresas con equipos de análisis de datos bien capacitados aún tienen que trabajar mucho. (Álvarez C.J., 2021).

Figura 1

Industria Conectada 4.0



Nota. En la figura se visualiza un claro ejemplo de la Industria 4.0. Tomado de Álvarez C. J., (2021).

La Industria 4.0 conecta los sistemas físicos con el mundo digital, de esta forma se monitorizan todo tipo de variables y el sistema tiene cierto grado de autonomía incluyendo la capacidad

de tomar decisiones. Esto es lo que impulsa a las empresas a integrar procesos de fabricación inteligente donde la intervención humana es mínima. (Almada L. 2016).

La integración vertical de sistemas de producción inteligentes se refiere al hecho de que las fábricas inteligentes, que son esencialmente el corazón de la Industria 4.0, no pueden operar de forma aislada. Es necesario construir una red de fábricas inteligentes, productos inteligentes y otros sistemas de producción inteligentes. La naturaleza de la red vertical se deriva del uso de un sistema de producción electrónico que permite que las fábricas y plantas de fabricación respondan de manera rápida y adecuada a variables como niveles de demanda, niveles de inventario, defectos de producción, etc., maquinaria y retrasos. Asimismo, la creación de redes y la integración también están asociadas con los departamentos de *marketing* y logística inteligente de una organización y sus servicios inteligentes, ya que el proceso de fabricación es individualizado, personalizado y exclusivo para el cliente. (Velásquez L., & Alba L., 2019).

La ingeniería completa en toda la cadena de valor esto significa que toda la cadena de valor industrial está sujeta a lo que se denomina ingeniería avanzada, todo el ciclo de vida del producto, desde la producción hasta la distribución al consumidor final. Para los componentes industriales, la calidad es el sello distintivo, por lo que el fabricante debe crear productos que cumplan con las expectativas del cliente. Industria 4.0 cubre todo el proceso de fabricación y el ciclo de vida del producto (Velásquez L., & Alba L. 2019).

Además de los factores anteriores, la tecnología más importante en la Industria 4.0 son las Tecnologías de la Información y las Comunicaciones (TIC). En Industria 4.0, las nuevas capacidades están surgiendo tan rápidamente que es difícil para las organizaciones industriales mantenerse al día. (Đuričin & Herceg, 2018).

Campos de aplicación

La Industria 4.0 está impulsada por tecnologías digitales que están teniendo un impacto devastador en los modelos y procesos comerciales corporativos y creando valor para los clientes. (Sjøbakk, & Đuričin & Herceg, & Salkin, et al., 2018).

Internet Industrial de las cosas (IIoT) es una red abierta y no determinista en la que las entidades inteligentes y los objetos virtuales autoorganizados pueden interactuar entre sí y pueden perseguir de forma independiente sus propios objetivos según el contexto, la situación o el entorno. (Sjøbakk, & Đuričin & Herceg, & Salkin, et al., 2018).

Internet de las cosas (IoT)

El concepto de Internet de las Cosas fue introducido en 1999 por el ingeniero Bill Joy, quien es el responsable de descubrir su gran utilidad y todos los métodos a los que se puede aplicar esta herramienta. (Terán O. 2017). El Internet de las Cosas es el concepto de red de intercambio de información y comunicación a través de Internet, para lograr una gestión inteligente, cuyo propósito es permitir que toda la interfaz pueda comunicarse entre sí en cualquier momento, comunicándose con otros hilos e incluso controlarlos. (Pajares A. 2016).

Se puede pensar en IoT como una combinación de sensores y actuadores capaces de proporcionar y recibir información digitalizada y colocarla en redes bidireccionales capaces de transmitir todos los datos utilizados por un solo dispositivo para una gran cantidad de diferentes servicios y usuarios finales. (Ch. Perera, 2014).

La arquitectura del sistema IoT se puede dividir en cuatro capas: la capa de descubrimiento de objetos, la capa de intercambio de datos, la capa de integración de información y la capa de servicio de aplicaciones. (Ma HD. 2011).

IOT como la red de redes

Hoy en día, Internet de las cosas consiste en una colección suelta de diferentes redes con diferentes propósitos. Por ejemplo, los automóviles de hoy en día tienen varias redes para controlar el funcionamiento del motor, las medidas de seguridad, los sistemas de comunicación y más. De manera similar, los edificios comerciales y residenciales tienen diferentes sistemas de control para calefacción, ventilación, aire acondicionado, teléfono, seguridad e iluminación. (D. Evans, 2011).

Elementos del IOT

El Internet de las cosas tiene áreas de aplicación que juegan un papel importante al influir en las características físicas del lugar de trabajo en las industrias que adoptan el Internet de las cosas, para así cambiar la forma en que interactúan con los dispositivos para adoptarlos e implementarlos con éxito, lo que plantea una serie de retos importantes para las personas y las organizaciones. (Goumagias, N., Whalley, J., Dilaver, O. Cunningham, J., 2021).

La Industria 4.0 para automatización, análisis, robótica y sensores está relacionada con varias áreas de aplicación relacionadas con Internet de las cosas, incluida la satisfacción de habilidades y necesidades, y el uso de herramientas de visualización de trabajos, es una característica importante para respaldar la toma de decisiones en Industria 4.0, teniendo en cuenta que hay estudios en la literatura que muestran que el Internet de las Cosas a menudo puede ser impulsado por intervenciones tecnológicas genéricas en lugar de innovación para modelos de negocios. (Matsuo, K. Barolli, L., 2020).

Las empresas que integren el Internet de las Cosas pueden potenciar su liderazgo para impulsar el cambio hacia la sostenibilidad, más allá del impacto positivo en la vida de las personas, por beneficios y adaptabilidad de la tecnología liderada por el Internet de las Cosas, es fundamental tener suficiente conocimiento y comprensión de Internet de las cosas para ir más allá de las desventajas de la tecnología

y, por lo tanto, tener un impacto positivo en la motivación de las personas para usar Internet de las cosas. (Kim, K.J. Wang, S., 2021).

Las herramientas que permiten una fácil visualización e interpretación de la información y deben diseñarse para adaptarse a diferentes aplicaciones y dispositivos. (In , L., & Kyoochun , L. 2015).

Objetivo del IoT.

La investigación en construcción para el Internet de las Cosas abarca una variedad de áreas cuyo objetivo siempre ha sido brindar soluciones y mejorar problemas que requieren el uso de tecnología y conectividad a Internet, y brindar conocimientos avanzados a la industria de la construcción, el uso integrado de sensores, la precisión y la construcción a pequeña escala como impulsores de las ganancias en sostenibilidad. (Oke, A.E., Arowoia, V.A., 2021).

Se espera que IoT sea ampliamente utilizado en la industria de la construcción, el uso de energía renovable de acuerdo con los servicios de IoT es esencial, y el desarrollo de IoT verde ahorra energía, junto con los profesionales de la construcción, los beneficios aparecen cuando las tecnologías funcionan basado en la mejora de la eficiencia inteligente en proyectos de construcción, las tendencias de la aplicación IoT-BIM en la creación de plataformas útiles en la transformación digital de la gestión, las operaciones y el mantenimiento de la energía, seguidas por la tendencia de utilizar el modelo de información de construcción (BIM) en el sistema basado en el uso sensores para la recogida de datos. (Sodring, T., Reinholdtsen, P. Massey, D., 2020).

Cloud computing

La computación en la nube nos permite brindar servicios informáticos en línea a los que podemos acceder desde cualquier lugar teniendo la autorización. (Kezherashvili B., 2018).

La “computación en la nube” es un nuevo modelo para proporcionar tecnología y servicios comerciales, que permite a los usuarios acceder a un conjunto unificado de servicios y satisfacer sus

necesidades comerciales, de manera flexible y adaptable, en caso de demanda máxima o trabajo inesperado, pagando únicamente por el consumo efectuado. (Kezherashvili B., 2018).

Muchos conceptos de computación en la nube se refieren a un nuevo estilo de computación dinámicamente escalable que, a través de la virtualización de recursos, como *hardware*, *software* y aplicaciones, puede brindar servicios a través de Internet. (Furht & Escalante, 2010).

La computación en la nube también se conoce como el nuevo modelo de recursos informáticos y de datos que se ejecuta en la plataforma de la nube. Aquí encontrará una gama de centros de datos privados y trabajos de mantenimiento de terceros. (Antonopoulos & Gillam, 2010).

Características del cloud computing

- Costes más bajos, los servicios en la nube son cada vez más eficientes y, por lo tanto, su uso es más productivo.
- Fácil de usar, los usuarios de la nube no tienen que preocuparse por las licencias de *software* o *hardware* para implementar el servicio, el proveedor se encarga de todo.
- Administración TIC subcontratada, El departamento de servicios TIC es un departamento externo, lo que permite a la empresa centrarse en la gestión de su negocio.
- Mantenimiento y actualización simplificados, el sistema está centralizado y se facilitan las actualizaciones, lo que permite un fácil acceso a las últimas versiones. (Sosinsky, B.2011).

Modelos de servicio del cloud computing

IaaS (infraestructure as a service). Esta categoría proporciona servicios de infraestructura. Entre ellos, la distribución de recursos informáticos y de almacenamiento, los cuales tienen un precio variable en función del consumo. (Antonopoulos N. y Gillam L., 2010).

PaaS (plataform as a service). Este servicio proporciona plataformas de desarrollo sin la adquisición de tecnología a un costo muy alto. El proveedor de servicios administra el *hardware* y el *software* de este modelo, y a los desarrolladores no les importa el rendimiento del *hardware*, y mucho menos las actualizaciones del sistema operativo, para todo lo que hace el proveedor. La oferta es un servicio relacionado con el rendimiento. Los proveedores de soluciones PaaS más populares son *Windows Azure* y *Google App Engine*. (Jamsa, 2012).

SaaS (Software as a Service). Esta es la forma más utilizada porque además de brindar servicios de *software*, también brinda la capacidad de almacenar información generada a partir del *software*. Las ventajas notables de este modelo son la facilidad de integración, el costo y la escalabilidad. Conocer incidentes de seguridad y mantener información almacenada en lugares físicos distintos a los espacios corporativos pueden ser considerados errores. Estos proveedores de servicios incluyen *Google Apps*, *TurboTax* y *QuickBooks*. (Jamsa, 2012).

Beneficios del cloud computing

Entre los principales beneficios de *Cloud Computing* podemos destacar los siguientes:

- Acelera el lanzamiento de nuevas aplicaciones
- Rompe fronteras y expande los negocios
- Promueve la participación de los clientes
- Impulsa a los empleados

- Optimiza procesos
- Cuenta con capacidad de almacenamiento ilimitada
- Respeta al medio ambiente

La rápida adopción de este servicio en la economía global y la capacidad de reducir los costos asociados con el diseño, fabricación y gestión de los bienes y servicios que brinda la empresa en general crean importantes ventajas y le permiten competir en el mercado global. (Mariscal Judith & Gil García Ramón, 2013).

Riesgos en entornos cloud computing

- A la hora de elegir un proveedor, se debe tener en cuenta su reputación, trayectoria y sostenibilidad. Este último es muy importante porque asegura que los servicios estarán disponibles y se podrá rastrear la información.
- Los proveedores de servicios son responsables de procesar la información, y el incumplimiento de los niveles de servicio acordados afectará negativamente la seguridad, la disponibilidad y, por lo tanto, el negocio.
- Puede haber confusión sobre la ubicación de la información, lo que puede causar demoras en la recuperación de la información.
- El acceso a información confidencial por parte de terceros compromete la seguridad de la información y puede constituir una amenaza para la protección de los derechos de propiedad intelectual.
- En las nubes públicas, la desventaja es la posibilidad de mezclar contenido informativo con contenido informativo de otros clientes de la nube o incluso de la competencia.

- En el caso de un desastre, es posible que la información no se encuentre de inmediato, por lo que los planes de recuperación y continuidad del negocio deben documentarse y probarse por completo. (ISACA, 2009).

Tipos de cloud computing

Existen diferentes tipos de nubes (*cloud computing*) según las necesidades del negocio, el modelo de servicios que se ofrecen y cómo se implementan. Dependiendo de dónde estén instaladas las aplicaciones y cómo puedan usarlas los clientes, tendremos nubes públicas, privadas o híbridas.

Nubes Públicas. La ventaja más obvia de las nubes públicas es la capacidad de procesar y almacenar sin instalar una máquina local, por lo que no tiene que invertir en inversión inicial o costos de mantenimiento en este sentido, pero sí tiene que pagar para usarlo. Uno de sus inconvenientes es el acceso a toda la información de terceras empresas y depende de servicios en línea.

Nubes Privadas. Sin embargo, en las nubes privadas, la plataforma está en las instalaciones del usuario y no suele prestar servicios a terceros. La ventaja de este tipo de nube, a diferencia de una nube pública, es como se encuentran los datos en la institución o empresa, lo que aporta mayor seguridad. Sin embargo, la desventaja es la inversión inicial en infraestructura física, sistemas de virtualización, ancho de banda y seguridad, lo que a su vez conduce a una pérdida de escalabilidad e inestabilidad de la plataforma, sin mencionar el costo y el mantenimiento que requiere. Esta mayor inversión dará como resultado un retorno de la inversión más lento. La ventaja más obvia de las nubes públicas es la capacidad de procesar y almacenar sin instalar una máquina local, por lo que no tiene ninguna inversión inicial ni costos de mantenimiento en este sentido, pero debe pagar para usarla. Uno de sus inconvenientes es el acceso a toda la información de terceras empresas y depende de servicios en línea.

Nubes Híbridas. Estas incluyen la combinación de aplicaciones locales y de nube pública.

También se puede considerar como una aplicación privada que se ve aumentada con los servicios de *Cloud Computing* y la infraestructura. Esto permite a las empresas mantener el control de sus aplicaciones principales, mientras aprovecha la computación en la nube donde tiene sentido. (Kezherashvili B., 2018).

Arquitectura de la nube

Una arquitectura de computación en la nube consiste en un conjunto de capas agrupadas para brindar funcionalidad al sistema, en este caso una arquitectura de computación en la nube es similar a una arquitectura de red, desde el nivel físico hasta el nivel de aplicación. De hecho, la computación en la nube utiliza protocolos similares a los que se utilizan en Internet como medio de comunicación, ya sea basado en la web o no.

La arquitectura genérica de *Cloud Computing* consta de las siguientes capas:

- Recursos físicos: en los que se encuentra elementos como servidores, para almacenar la información.
- Infraestructura: incluye *software* de plataforma como servicio.
- Plataforma: incluye componentes de aplicación como servicio.
- Aplicación: incluye servicios basados en *Web* y *software* como servicio. (Kezherashvili B., 2018).

Protocolo MQTT

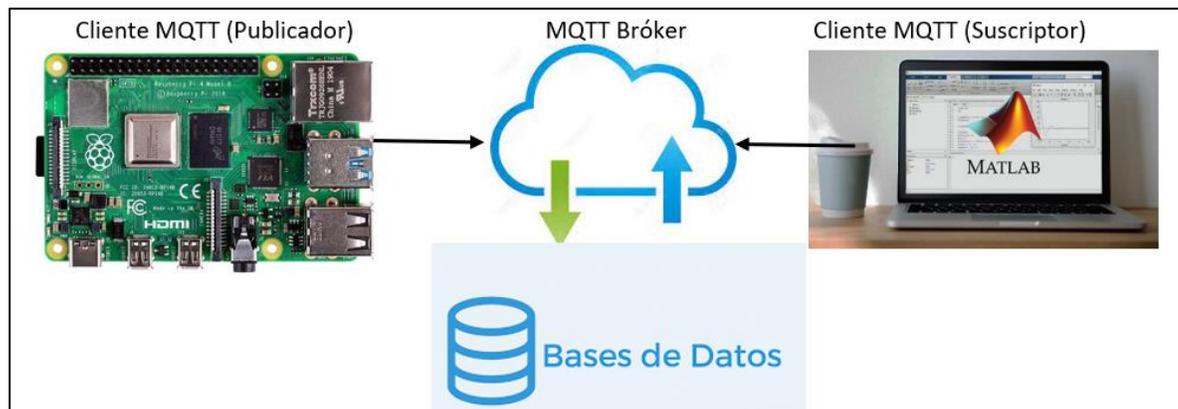
MQTT es un protocolo de mensajería diseñado para ser abierto, liviano y simple, ideal para entornos restringidos donde las redes son costosas o los recursos de memoria del dispositivo son limitados, se propaga a través de TCP-IP y hay tres calidades de servicio, donde el mensaje se enviará

como mucho una vez, al menos una vez o exactamente una vez, y esto asegura que el mensaje llegue a su destino y que su transmisión sea independiente de la descargar. (IBM & Eurotech, 2010).

MQTT trabaja sobre protocolos de red que permiten mantener comunicaciones bidireccionales ordenadas y asimétricas, entre sus características se encuentran la entrega de mensajes de uno a muchos, transmisión de mensajes independiente de su contenido, calidad de servicio en la entrega de mensajes, tráfico de red reducido y notificación a interesados de cortes de conexión extraordinarios. (mqttv3.1.1, 2014).

Figura 2

Funcionamiento del protocolo MQTT



Nota. Tomado de Borja Edwin, 2020.

Los mensajes en MQTT se publican en *topics*, no se requiere configuración, solo la publicación es suficiente, los *topics* se tratan como una jerarquía separada con una barra inclinada (/), por ejemplo: varios clientes pueden acceder a los mensajes en función de la jerarquía de acceso de la información de interés. (Borja E., 2020).

Arquitectura del sistema

La arquitectura de software consiste en la estructura de los componentes de un programa o sistema, sus interrelaciones y los principios y reglas que rigen su diseño y desarrollo a lo largo del tiempo. (Garlan, Perry, 1995).

Además de los algoritmos y las estructuras de datos computacionales, el diseño y especificación de la arquitectura del sistema global ha surgido como un nuevo tipo de problema. La estructura detallada incluye: la organización general y la estructura de control global, protocolos de comunicación, sincronización y acceso a datos; asignar funcionalidad a elementos de diseño; diseño físico, configuración de elementos de diseño, aspectos de escalabilidad y rendimiento, (Garlan, Shaw, 1993).

Componentes hardware del sistema

Los componentes de hardware a gestionar por un sistema operativo son la CPU, la memoria principal y los periféricos, incluyendo entre éstos a las memorias secundarias, (Brookshear, 2011).

Broker MQTT

El *broker* MQTT nos permite recibir y procesar mensajes de clientes y enviar mensajes a suscriptores de este hilo. Además, administro la red de tal manera que cuando el cliente envía mensajes constantemente al *broker*, obteniendo una respuesta instantánea de este corredor.

Dentro del mercado podemos encontrar una infinidad de alternativas entre las principales tenemos al *broker Mosquitto*, y al *broker Mosca*.

Mosca. Es un agente MQTT de Node.js que se puede usar como un servicio independiente o integrado en otra aplicación de *Node.js*. Este *broker* brinda una gran comodidad para ver el mensaje, aunque en el instalador debe agregar una herramienta llamada *bunyan* para ver el mensaje de depuración.

Mosquitto. Es un intermediario de mensajes MQTT, es liviano y adecuado para todos los dispositivos, desde PC hasta servidores completos. Descargarlo e instalarlo en *Linux* es simple porque el administrador de paquetes Debian lo posee. (García, 2017).

NodeRed

Es una herramienta de programación visual para conectar dispositivos, API y servicios de formas nuevas y sencillas. *NodeRed* comenzó a principios de 2013 como un proyecto paralelo de Nick O'Leary y Dave Conway-Jones del grupo de servicios de tecnología emergente de IBM. (Kumar, Maurya, Dwivedi, 2019).

Lo que comenzó como una prueba de concepto y manipulación de mapas entre temas de MQTT evolucionó rápidamente hasta convertirse en una herramienta más general que se podía expandir sencillamente en cualquier dirección. Sus principales características son:

Edición de flujo basado en un navegador. *NodeRed* proporciona un editor de transmisión con varios botones disponibles en el tablero y se puede implementar con un solo clic. Las funciones de script de *Java* incluyen una biblioteca que le permite guardar funciones, formularios o flujos usados para su reutilización.

Node.js. *NodeRed* está basado en el entorno de desarrollo de *Node.js*, por lo que el tiempo de ejecución es suave, lo cual le permite aprovechar su modelo basado en eventos. La excelente calidad que posee lo hace ideal para ejecutarse en dispositivos de bajo costo como *Raspberry Pi* o en la nube. (Kodali,Soratkal, 2017).

Matriz de confusión

La Matriz es una herramienta para visualizar el rendimiento de los algoritmos utilizados en el aprendizaje supervisado. Esta herramienta se puede aplicar a problemas de n tipos. Es una matriz $n \times n$. Esta herramienta se utiliza para mostrar claramente cuando se ejecuta un modelo educativo, cuando

una capa se confunde con otra. Por eso te permite trabajar con diferentes tipos de errores por separado. (Ramirez J., 2018).

Contiene el número de estados categorizados correctamente y categorizados incorrectamente. Los estados bien clasificados caen en la diagonal, porque el grupo esperado y el grupo real son el mismo. Los elementos fuera del diámetro representan casos de clasificación errónea, (Malhotra, 2004). La siguiente figura muestra cómo poner los datos en una matriz de confusión.

Figura 3

Distribución matriz de confusión

		Predicción	
		Positivos	Negativos
Observación	Positivos	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Nota. En la figura se muestra la representación de la matriz de confusión, donde la diagonal principal se encuentra la suma de las predicciones válidas. Los elementos fuera de la diagonal reflejan el error del clasificador, sin embargo, cuando las diferentes clases están demasiado desequilibradas, la forma del clasificador puede volverse algo inútil. Cuando la precisión del algoritmo se mide de esta manera, no se hace distinción entre falsos positivos y falsos negativos y se supone que ambos tienen la misma importancia. Tomado de Ramirez J., 2018.

Falsos negativos (FN). En el caso de la Figura 2.3, los falsos positivos son todas las pruebas que el SVM considera una muestra negativa, pero en realidad pertenecen a una muestra positiva.

Falsos positivos (FP). Como se muestra en la Figura 2.3, un resultado falso negativo es el conjunto de pruebas que se describen como una muestra positiva, pero que en realidad pertenecen a la muestra negativa.

Verdaderos positivos (VP). Estos se denominan verdaderos positivos de un conjunto de prueba exitoso, en el caso de la ilustración, son positivos durante la configuración de SVM y en el modelo real.

Verdaderos Negativos (VN). El conjunto de pruebas exitosas se llama verdadero negativo. En el caso de la Figura 2.3, todo esto, en contraste con la verdadera positividad, la respuesta de aprendizaje de la SVM es negativa, como en el modelo real.

Accuracy (Exactitud). La exactitud se expresa como la relación entre el número de unidades de muestra clasificadas correctamente (los elementos diagonales de la matriz de confusión) y el número total de elementos que forman parte del mapa.

Sensibilidad. La sensibilidad se puede definir como la capacidad de un sistema de clasificación para generar datos positivos para valores que ya pertenecen a esa categoría. El valor que puede tomar la sensibilidad es de 0 a 1 o de 0 a 100%, es decir, cuanto mayor sea el valor de sensibilidad, mejor será la capacidad de detección de casos positivos en esta prueba.

Especificidad. A diferencia de la sensibilidad, la especificidad se puede definir como la capacidad de un sistema de clasificación para negar todos los valores que ya pertenecen a la categoría negativa. Su valor varía de la misma forma que en el caso de la sensibilidad entre 0 y 1 o 0 y 100%, indicando que, a mayor valor de especificidad, mejor capacidad del sistema para clasificar valores negativos de la prueba. (Ramirez J., 2018).

Capítulo III

Metodología

El desarrollo de algoritmos se basa en datos que describen un sistema en una variedad de condiciones saludables y defectuosas. Los datos puros se procesan previamente para darles un formato desde el cual se puede extraer indicadores de condición.

Los indicadores de condición son características que ayudan a distinguir las condiciones sanas de las defectuosas, las características extraídas se utilizan para entrenar un modelo de aprendizaje que pueda: detectar anomalías, clasificar diferentes tipos de fallos o estimar la vida útil restante de una máquina. Una vez obtenido el algoritmo, éste se integra al sistema de supervisión y mantenimiento de máquinas.

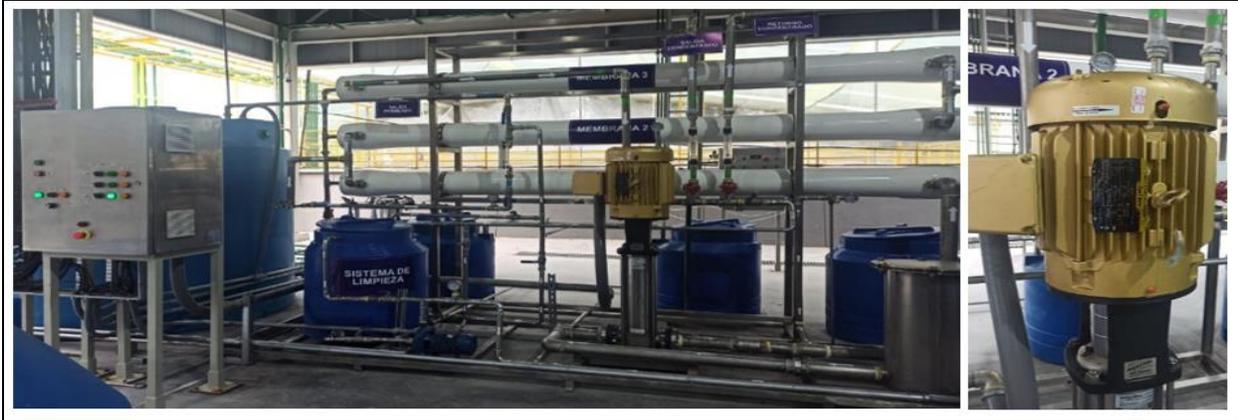
Descripción del proceso

El proceso de sistema de ósmosis inversa, para el cual se desarrolla el presente sistema, es parte de una planta de tratamiento de agua, éste recibe el agua a tratar, después de pasar un proceso de ultrafiltración y en la salida se tiene como resultado el agua de permeado que puede ser enviada a descargarse una vez que cumpla con los parámetros ambientales y el agua de concentrado que es enviada a ser procesada nuevamente.

En la figura 4 se puede observar una de las fallas más comunes presentadas por una bomba centrífuga dentro de un sistema de sistema de ósmosis inversa.

Figura 4

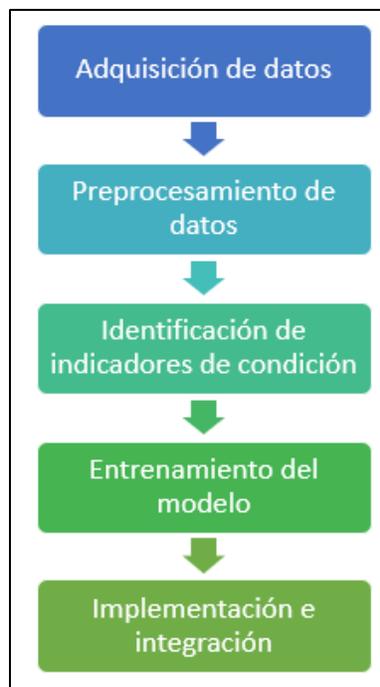
Sistema de ósmosis inversa



Método de desarrollo de un sistema de mantenimiento predictivo

Figura 5

Método de desarrollo de un sistema de mantenimiento predictivo

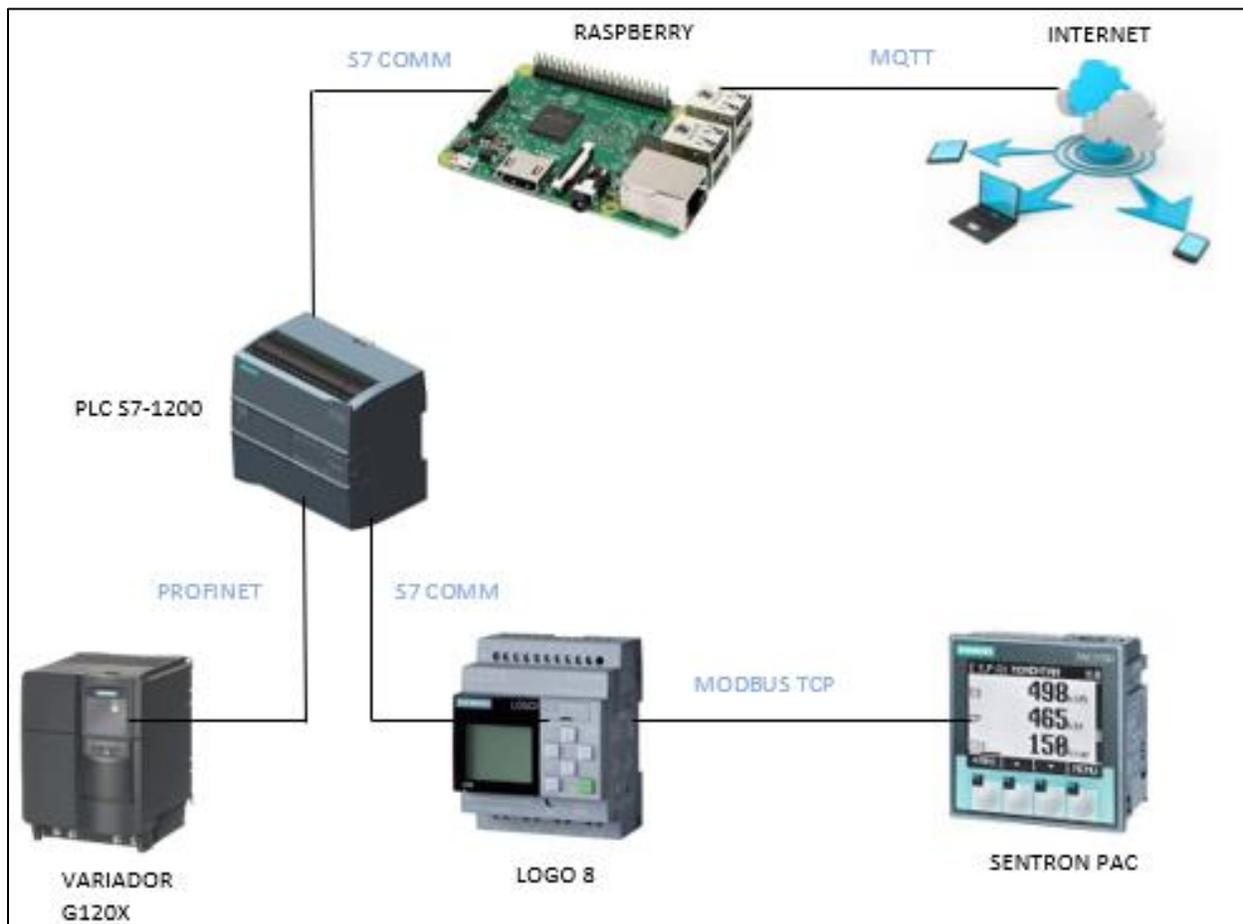


Nota. En la figura se muestra los pasos para el desarrollo de un sistema de mantenimiento predictivo.

Adquisición de datos

Figura 6

Arquitectura de red



Comunicación entre analizador de energía y módulo Logo 8

El *software* utilizado en esta etapa es *Logo Soft*, el cual mediante el protocolo *Modbus TCP*, permite adquirir los registros de datos desde el analizador de energía, este dispositivo permite obtener el valor de los parámetros eléctricos generales del sistema, como: tensión, intensidad, potencia consumida etc.

Comunicación entre módulo Logo 8 y PLC

El *software* utilizado en esta etapa es *Logo Soft* y TIA Portal, la comunicación se establece mediante el protocolo S7, éste protocolo permite enviar y recibir información entre controladores lógicos Siemens, accediendo directamente a las bases de datos de cada controlador.

Comunicación entre Variador y PLC

En esta etapa se utilizó TIA Portal, la comunicación se establece mediante el protocolo Profinet, éste protocolo permite enviar y recibir información entre el PLC y el Variador.

Comunicación entre PLC y raspberry

Mediante TIA Portal y NodeRed, se establece la comunicación mediante el protocolo S7, éste protocolo permite enviar y recibir información entre los bloques de datos del PLC y la plataforma *NodeRed* alojada en la *Raspberry*.

Comunicación entre raspberry y servidor web

Con el *software NodeRed*, se establece la comunicación mediante el protocolo MQTT, éste protocolo permite que los clientes se conecten con un servidor central denominado *broker*.

Almacenamiento de datos en MySQL

Se utiliza el *software NodeRed*, y el motor de datos *MySQL* para guardar los valores.

Desarrollo del sistema

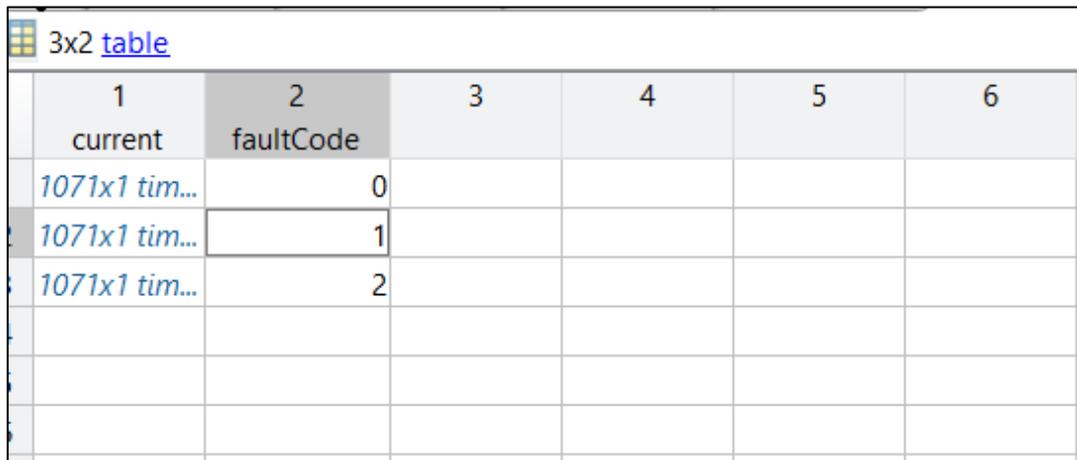
En esta etapa se utiliza el *software Matlab* y las herramientas para el desarrollo de algoritmos de mantenimiento predictivo las cuales son: *Diagnostic Feature Designer* y *Classification Learner*, la primera se usa durante el preprocesamiento de datos y en la identificación de indicadores de condición y la segunda para el entrenamiento del modelo de predicción.

Preprocesamiento de datos

En la figura 7 se muestra el formato que deben tener los datos cuando se ingresan a Matlab, para iniciar con el procesamiento a través del toolbox “DiagnosticFeatureDesigner”.

Figura 7

Ingreso de datos en Matlab



The image shows a screenshot of a Matlab table with 3 rows and 2 columns. The columns are labeled '1' and '2', with sub-labels 'current' and 'faultCode' respectively. The first three rows contain data: the first row has '1071x1 tim...' under '1' and '0' under '2'; the second row has '1071x1 tim...' under '1' and '1' under '2'; the third row has '1071x1 tim...' under '1' and '2' under '2'. The remaining three rows are empty.

1	2	3	4	5	6
current	faultCode				
1071x1 tim...	0				
1071x1 tim...	1				
1071x1 tim...	2				

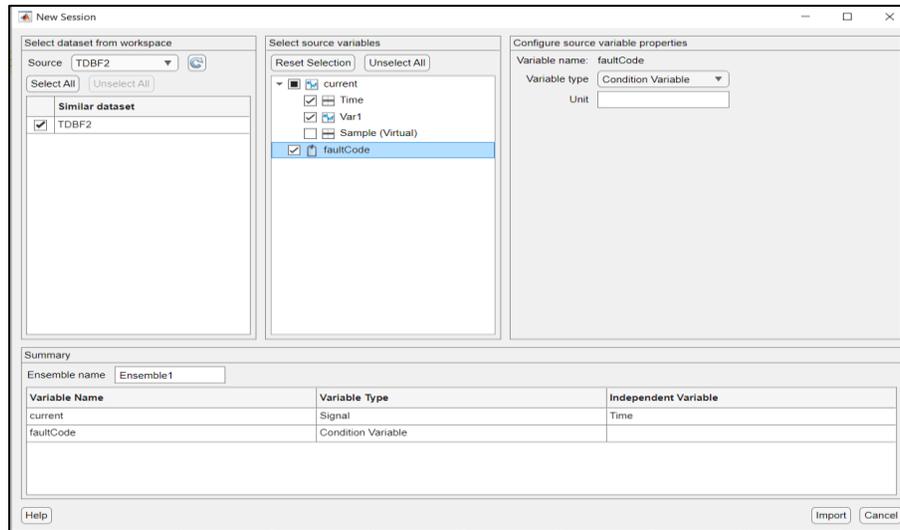
Los datos de la columna 1 (*current*) contiene los datos de corriente agrupados de acuerdo con el error que se produce, a los cuales se les asigna una codificación:

- *faultCode* = 0: El funcionamiento del sistema es normal.
- *faultCode* = 1: El código hace referencia a que las membranas del equipo se encuentran taponadas y la salida está obstruida.
- *faultCode* = 2: El código hace referencia a que la entrada al sistema está obstruida o el filtro se encuentra saturado.

Al abrir el toolbox “DiagnosticFeatureDesigner” se ingresa la variable declarada, luego se selecciona las variables con las que se trabajará y se asignará a la columna “*faultCode*” como variable condicional, finalmente se importa los datos, como se muestra en la figura 8.

Figura 8

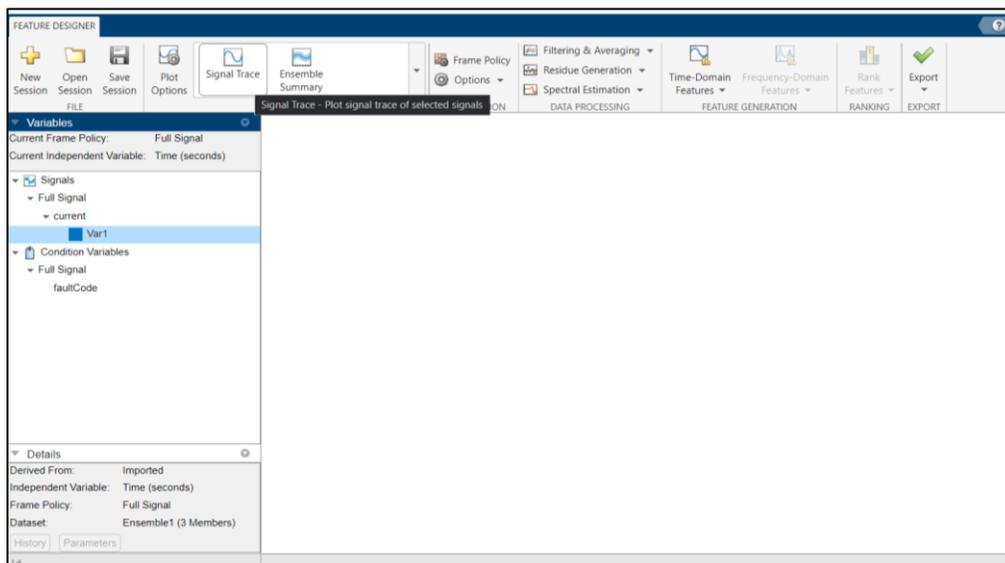
Asistente de importación de datos



Se selecciona “Signal Trace” para representar gráficamente los datos ingresados, como se muestra en la figura 9.

Figura 9

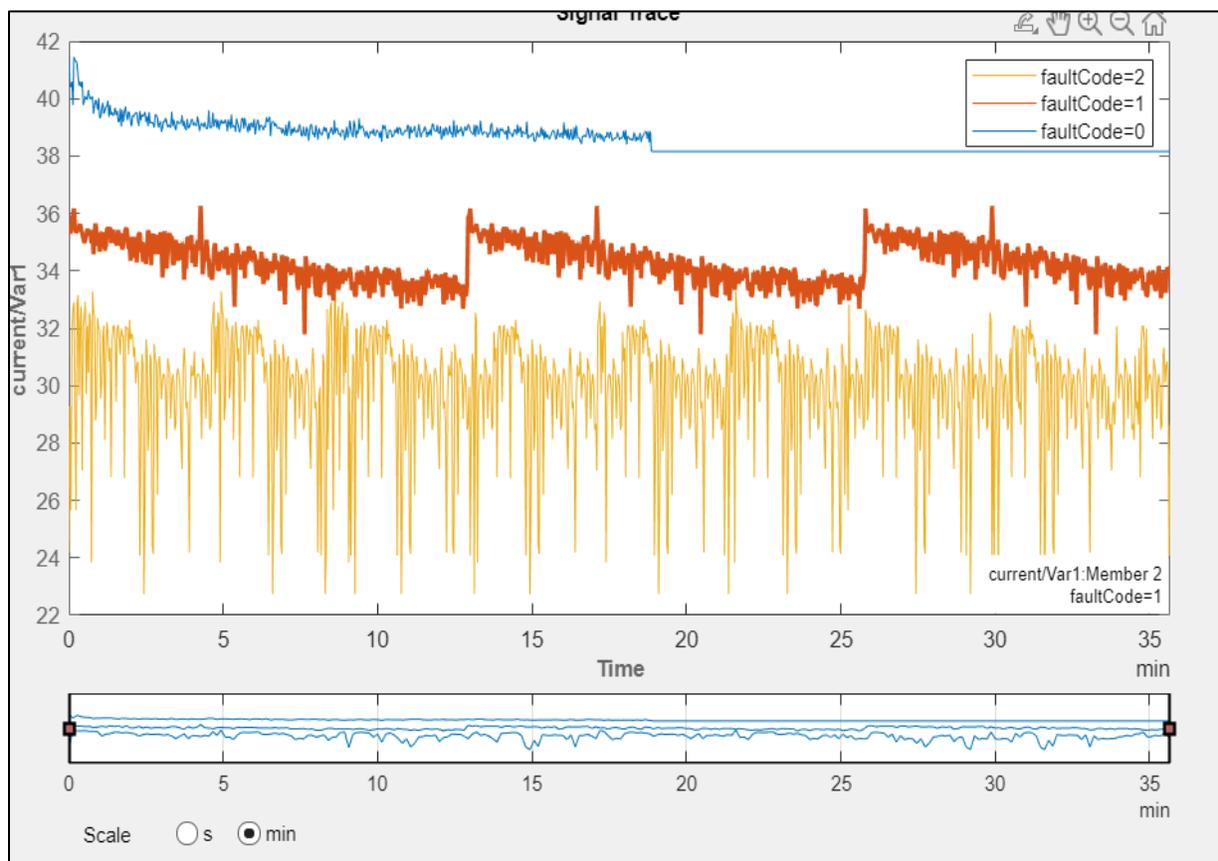
Interfaz de la herramienta Feature Designer en Matlab



Dentro de la interfaz donde se presenta una gráfica de los datos ingresados agrupados por el código de falla, de tal forma que se pueda diferenciar los datos, como se muestra en la figura 10.

Figura 10

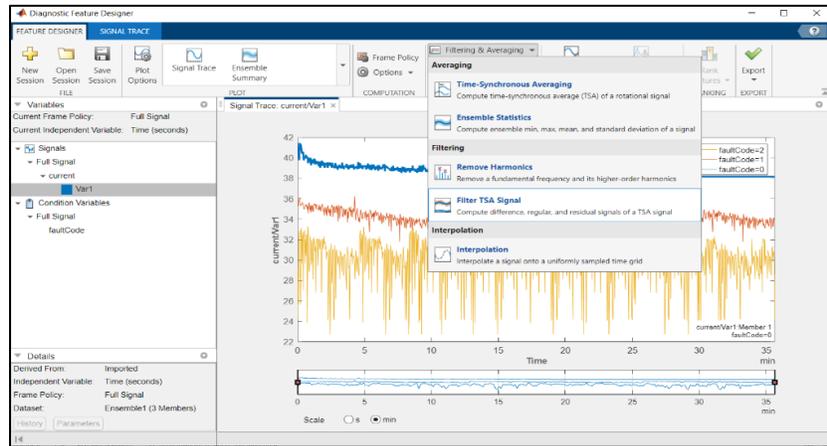
Representación gráfica de los datos puros



Los datos ingresados muestran un alto nivel de ruido por lo que deben ser filtrados, esto se realiza para incrementar la probabilidad de encontrar patrones y estos sean más fáciles de identificar durante el proceso de adaptación de la información para el entrenamiento, los resultados de este procesamiento se muestran en la figura 11

Figura 11

Aplicación de filtro en el conjunto de datos

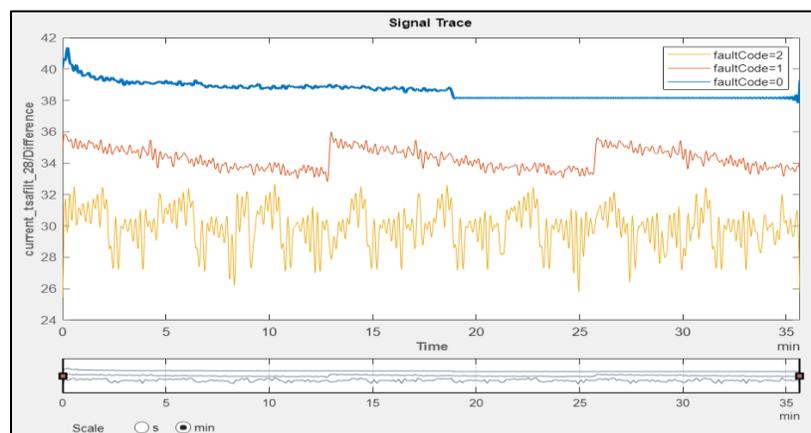


Después de aplicar el filtro se logran eliminar datos atípicos como se muestra en la figura 12.

El filtro utilizado para la etapa de preprocesamiento dentro de MATLAB es el promedio sincrónico de tiempo (TSA), el cual devuelve un promedio sincronizado en el tiempo de una señal, es una herramienta algorítmica esencial para determinar la condición del equipo rotativo, debido a su eficiencia en la reducción de ruido en las señales características de estos equipos.

Figura 12

Representación gráfica de los datos filtrados

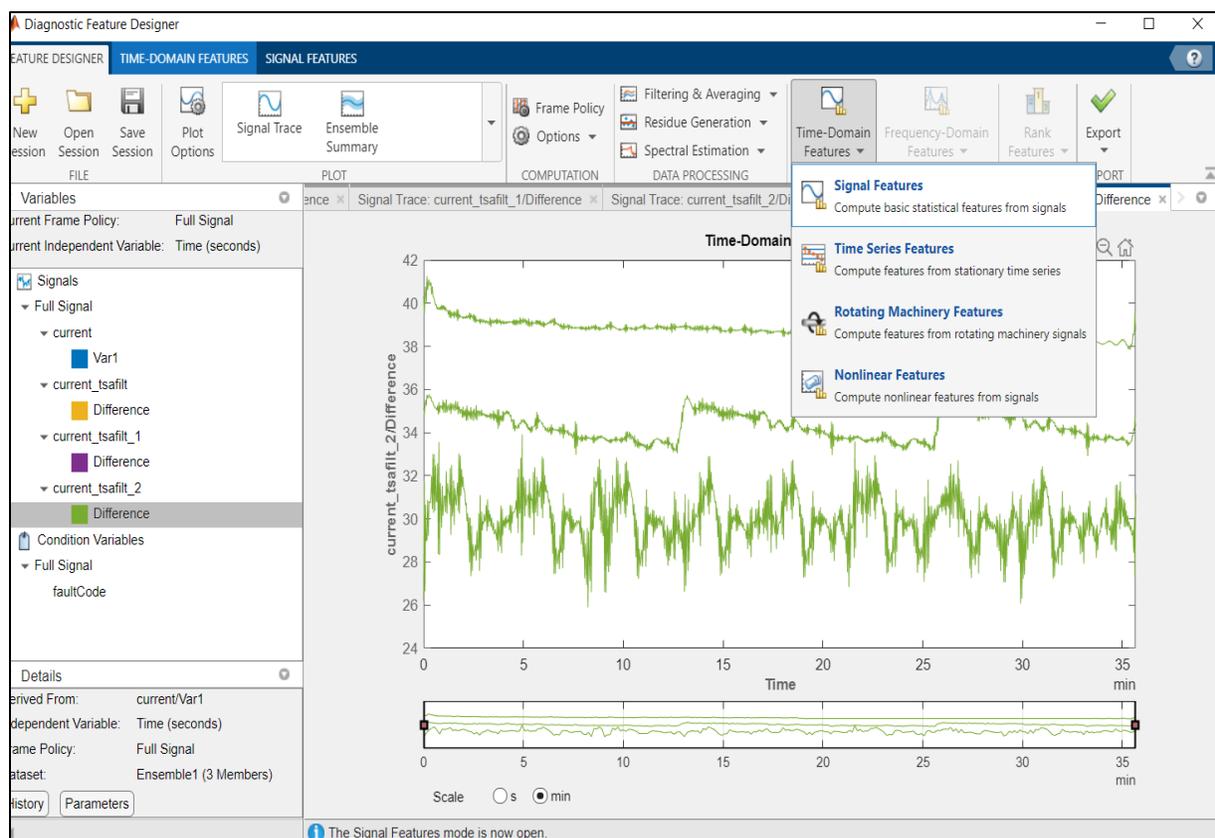


Identificación de indicadores de condición

Para identificar los patrones de comportamiento del sistema, se requiere extraer características tanto en el dominio del tiempo y en el dominio de la frecuencia, a continuación, se muestra el procesamiento realizado con los datos en función del tiempo, como se muestra en la figura 13.

Figura 13

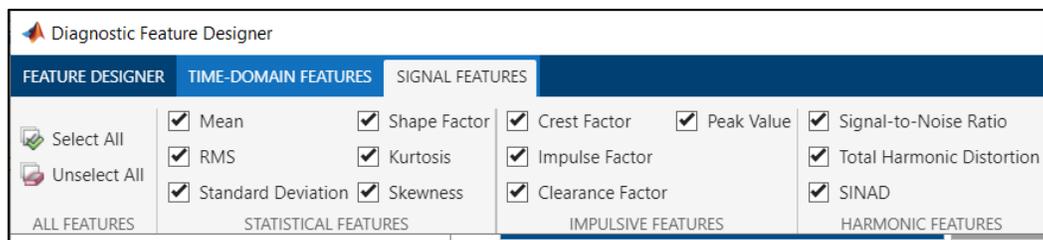
Procesamiento de datos en función del tiempo



Cuando aparezca el menú de configuración se debe marcar todas las opciones y aplicar los cambios, como se muestra en la figura 14.

Figura 14

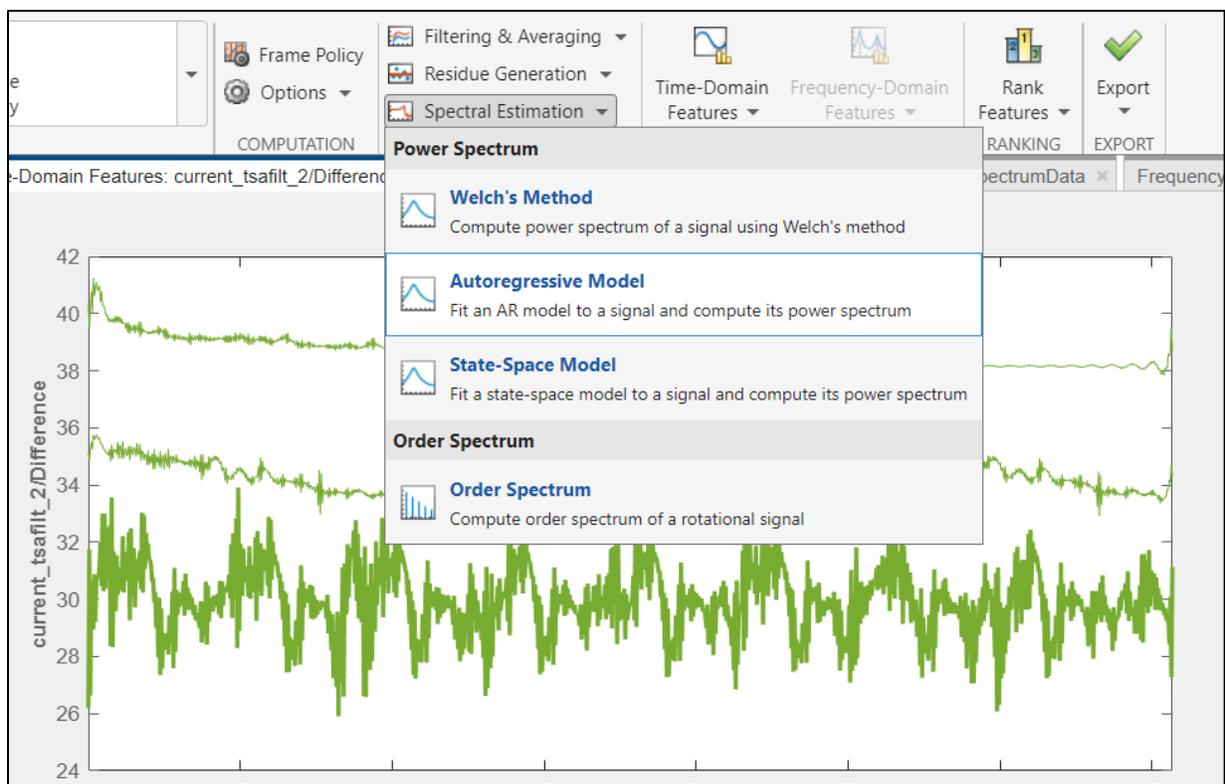
Menú de configuración en el dominio del tiempo



Para extraer las características en el dominio de la frecuencia se realiza una estimación de espectro seleccionando un modelo auto regresivo como se muestra en la figura 15, se usa este enfoque para reducir cualquier sobreajuste que exista en la señal de datos.

Figura 15

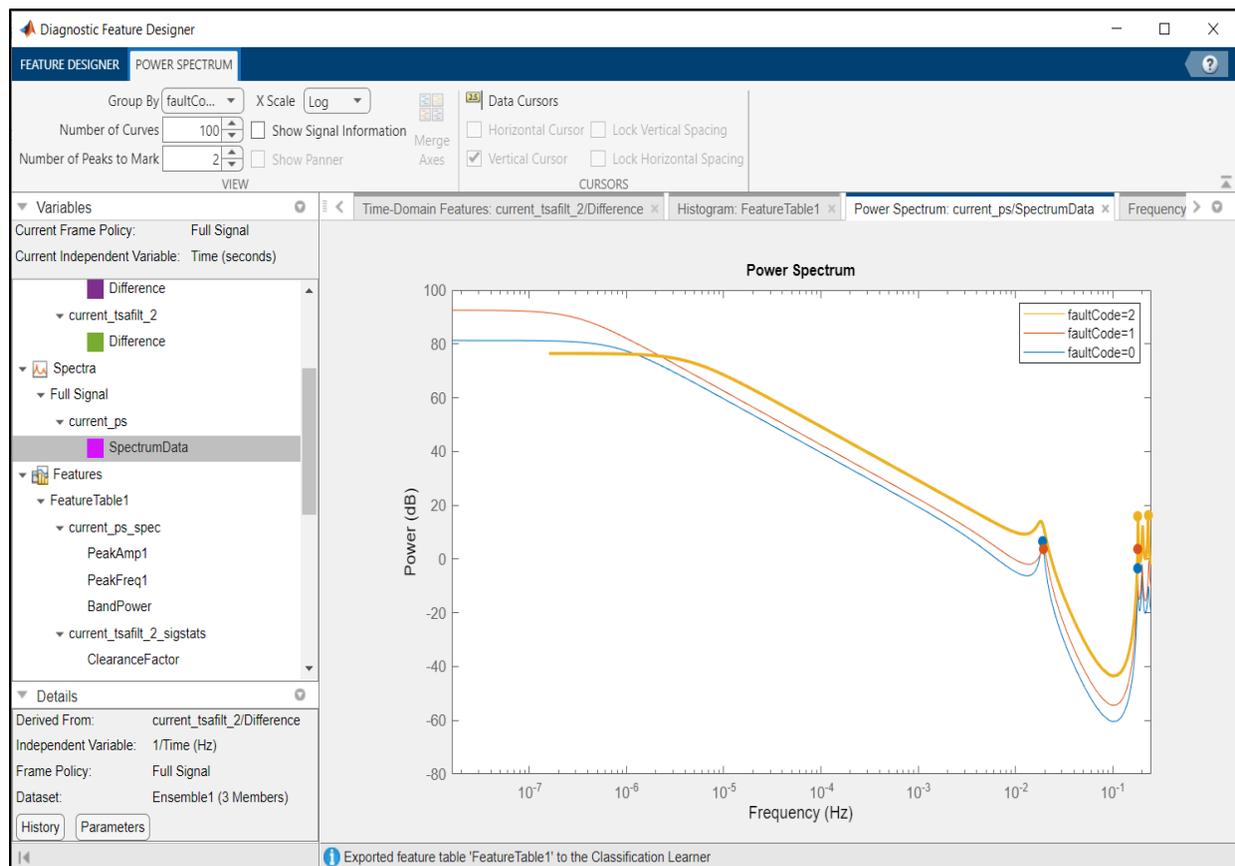
Ventana de configuración de modelo



Finalmente se obtiene el modelo auto regresivo de espectro en función de la frecuencia, como se muestra en la figura 16, esto se usa para extraer las características necesarias para el entrenamiento del sistema.

Figura 16

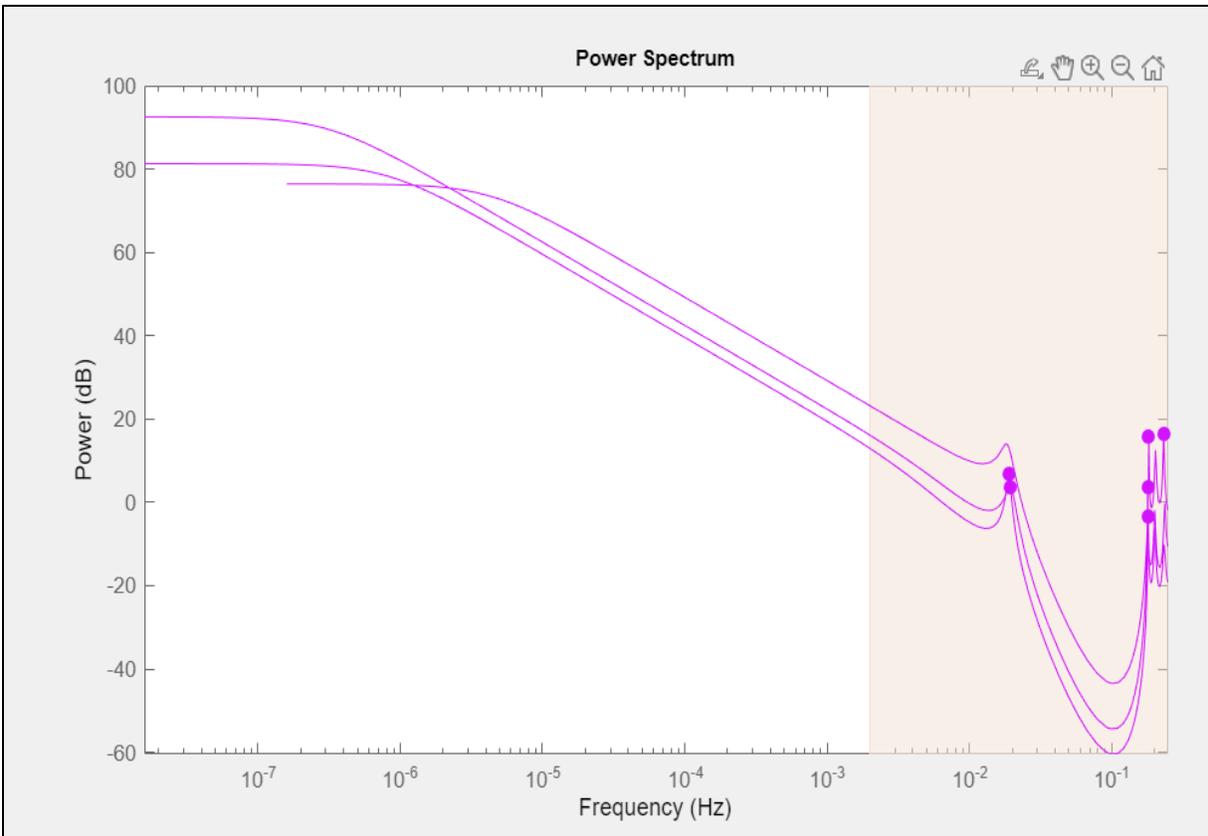
Modelo auto regresivo de espectro en función de la frecuencia



Para obtener las características del sistema se debe configurar los parámetros y discriminar los valores a ser procesados, en este caso los datos de interés se encuentran en el rango de 0.002 a 0.25 Hz, como se muestra en la figura 17.

Figura 17

Configuración de modelo auto regresivo en un rango de frecuencia



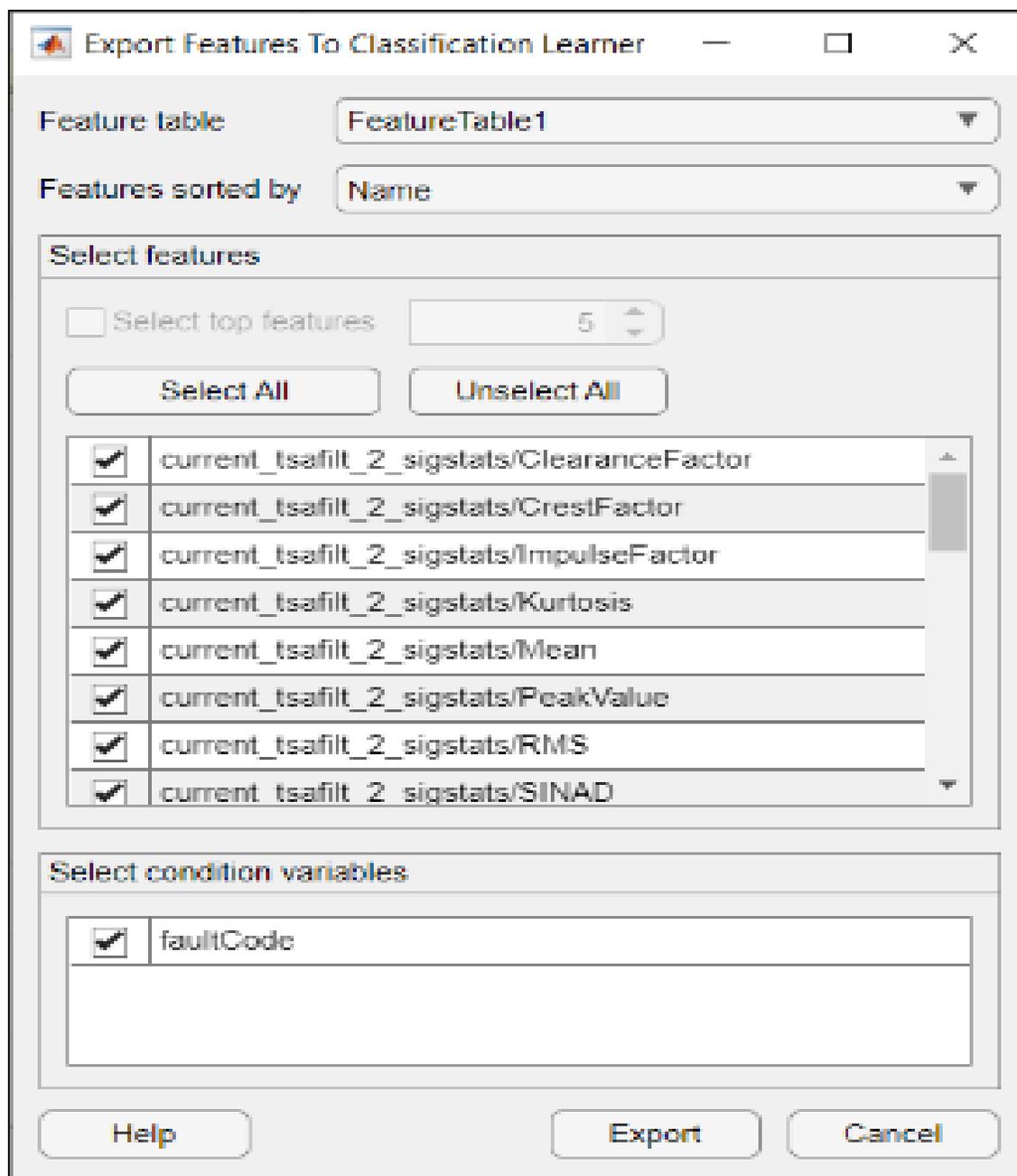
Una vez extraídos los parámetros en función de la frecuencia, la información se exporta para ser procesada posteriormente con la herramienta de *Matlab* "Classification Learner".

Entrenamiento del modelo

Dentro de la herramienta *Classification Learner* se abre una ventana donde se selecciona todas las características extraídas en la etapa anterior en el dominio del tiempo y la frecuencia, como se muestra en la figura 18.

Figura 18

Ventana de selección de características



Posteriormente se despliega un asistente que solicita la confirmación de los datos que se usarán para iniciar el entrenamiento, como se muestra en la figura 19.

Figura 19

Ventana de confirmación de datos seleccionados

Data set

Data Set Variable: FeatureTable1 (3x14 table)

Response: current_tsafilt_2_sigstats/THD (double, -6.0894...)

Predictors

	Name	Type	Range
<input type="checkbox"/>	faultCode	double	0 .. 2
<input checked="" type="checkbox"/>	current_tsafilt_2_sigstats/ClearanceFactor	double	1.04298 .. 1.13325
<input checked="" type="checkbox"/>	current_tsafilt_2_sigstats/CrestFactor	double	1.04271 .. 1.13179
<input checked="" type="checkbox"/>	current_tsafilt_2_sigstats/ImpulseFactor	double	1.04289 .. 1.13276
<input checked="" type="checkbox"/>	current_tsafilt_2_sigstats/Kurtosis	double	1.94917 .. 5.54332
<input checked="" type="checkbox"/>	current_tsafilt_2_sigstats/Mean	double	29.9351 .. 38.6105

Add All Remove All

[How to prepare data](#)

Validation

Validation Scheme: Cross-Validation

Protects against overfitting. For data not set aside for testing, the app partitions the data into folds and estimates the accuracy on each fold.

Cross-validation folds: 3

[Read about validation](#)

Test

Set aside a test data set

Percent set aside: 0

Use a test set to evaluate model performance after tuning and training models. To import a separate test set instead of partitioning the current data set, use the Test Data button after starting an app session.

[Read about test data](#)

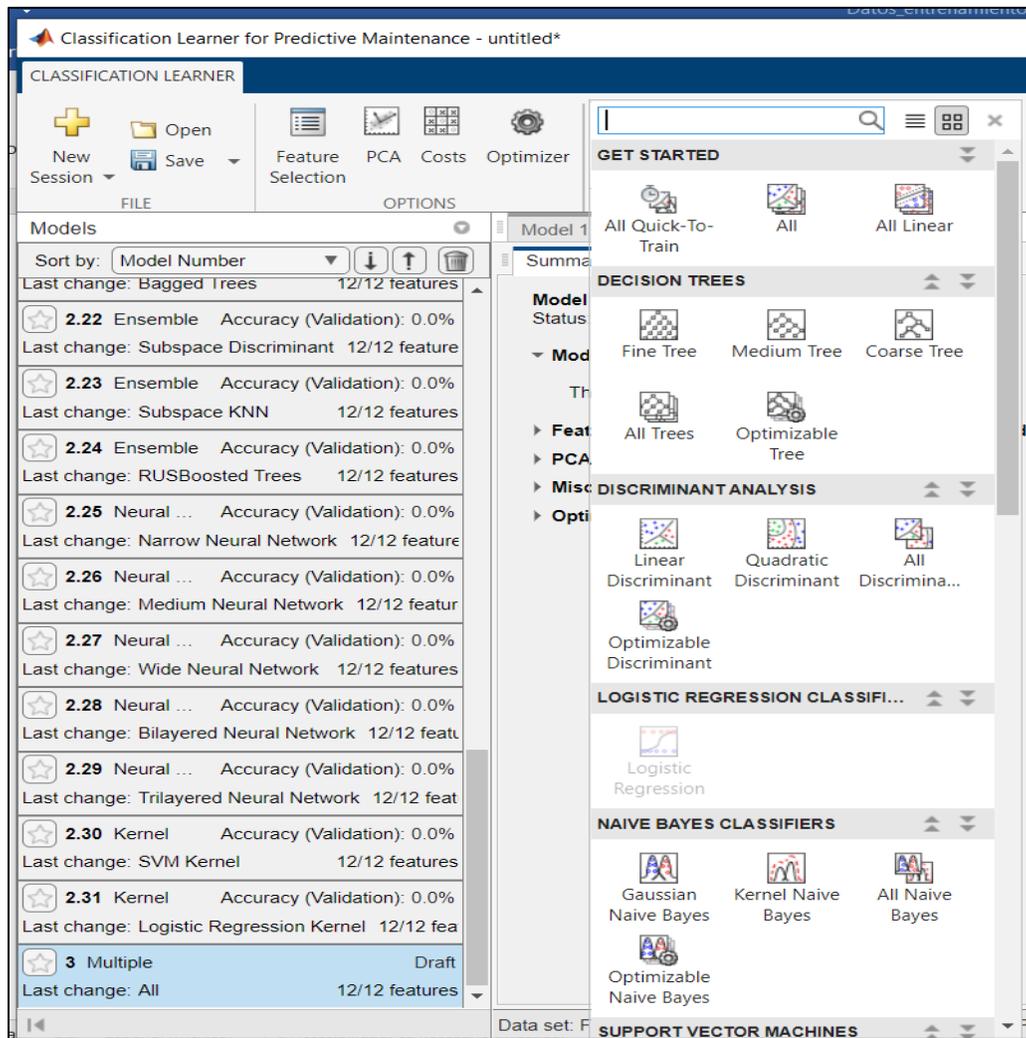
⚠ Response variable is numeric. Distinct values will be interpreted as class labels.

Start Session Cancel

Al iniciar la sesión se abre la interfaz de la herramienta de *Matlab*, *Classification Learner for Predictive Maintenance*, donde se encuentra un menú que despliega todos los modelos de entrenamiento disponibles como se muestra en la figura 20, en el cual se escoge la opción seleccionar todos, con el fin de que la herramienta encuentre el mejor modelo de entrenamiento para el sistema.

Figura 20

Ventana de la herramienta Classification Learner for Predictive Maintenance

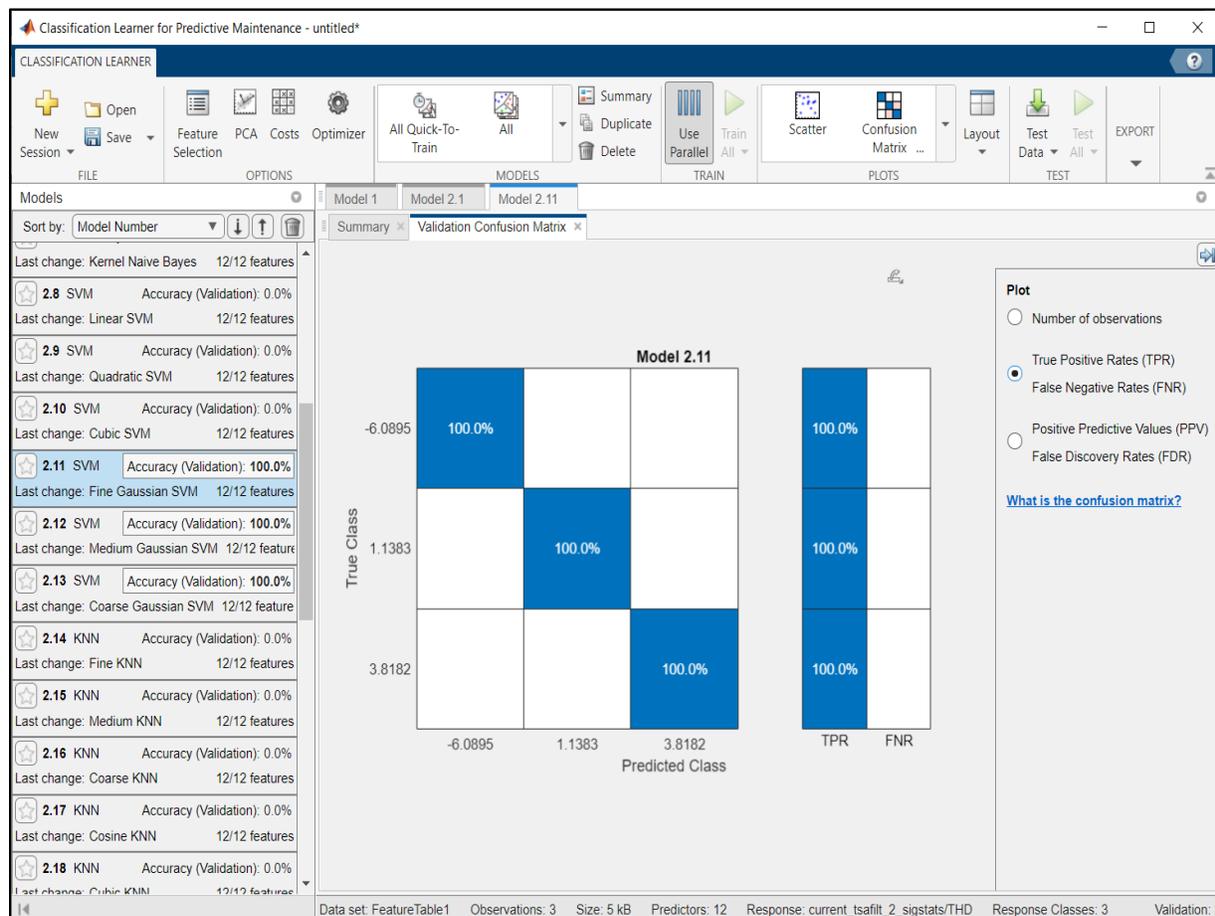


Quando el sistema finaliza la etapa de aprendizaje muestra los resultados obtenidos de cada modelo de aprendizaje seleccionado en la etapa anterior, y su porcentaje de precisión, finalmente se debe seleccionar el modelo con el porcentaje más alto, en los resultados obtenidos se obtienen tres modelos que cumplen con un porcentaje de precisión del 100%, estos son: *Fine Gaussian SVM*, *Medium Gaussian SVM* y *Coarse Gaussian SVM*, en este caso se selecciona *Medium Gaussian SVM*, debido a que

es el modelo más apropiado para sistemas de equipos rotativos, presentando el mejor desempeño en cuanto a confiabilidad y precisión según, (Orlowska T. & Wolkiewicz M., 2021).

Figura 21

Resultados de aprendizaje del modelo



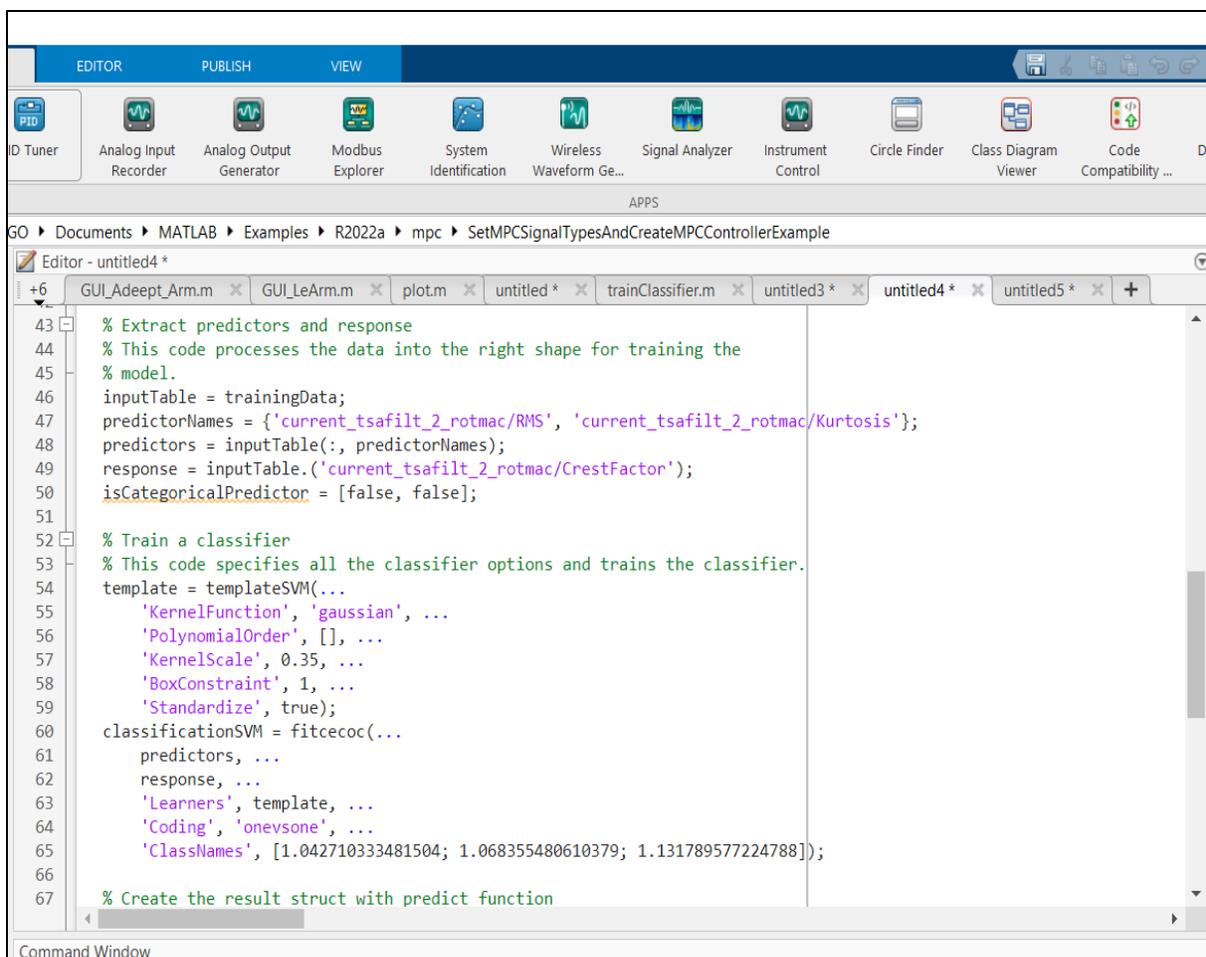
Se obtiene la matriz de confusión que permite visualizar los resultados del aprendizaje presentando el porcentaje de acierto entre el modelo real y el modelo de predicción, como se puede apreciar en la figura 21.

Implementación e integración

En la figura 22 se muestra el modelo obtenido es exportado desde la herramienta de Matlab, para posteriormente ser integrado en un bloque de función dentro del sistema de predicción en NodeRed.

Figura 22

Modelo obtenido en Matlab



```
43 % Extract predictors and response
44 % This code processes the data into the right shape for training the
45 % model.
46 inputTable = trainingData;
47 predictorNames = {'current_tsafilt_2_rotmac/RMS', 'current_tsafilt_2_rotmac/Kurtosis'};
48 predictors = inputTable(:, predictorNames);
49 response = inputTable('current_tsafilt_2_rotmac/CrestFactor');
50 isCategoricalPredictor = [false, false];
51
52 % Train a classifier
53 % This code specifies all the classifier options and trains the classifier.
54 template = templateSVM(...
55     'KernelFunction', 'gaussian', ...
56     'PolynomialOrder', [], ...
57     'KernelScale', 0.35, ...
58     'BoxConstraint', 1, ...
59     'Standardize', true);
60 classificationSVM = fitcecoc(...
61     predictors, ...
62     response, ...
63     'Learners', template, ...
64     'Coding', 'onesone', ...
65     'ClassNames', [1.042710333481504; 1.068355480610379; 1.131789577224788]);
66
67 % Create the result struct with predict function
```

Capítulo IV

Resultados

Interfaz de sistema de supervisión

La interfaz del sistema de supervisión se desarrolla en la plataforma *NodeRed* alojada en el servidor web, en donde se presenta en tiempo real los valores de: tensión de entrada, corriente consumida por el equipo, velocidad, torque, tensión de salida y temperatura, además se muestran gráficas que contienen el historial de los parámetros de voltaje, corriente y temperatura.

Se muestran también los resultados del análisis de los datos en tiempo real, el modelo generado evalúa los datos y da como resultado la codificación de fallo.

Placa del motor utilizado en el presente proyecto de investigación

Figura 23

Placa de motor Baldor

BALDOR . RELIANCE						
Super-E Motor						
CAT NO	VEM4107T		CUST	P/N	ENCL	TEFC
SPEC	09F909T577G1	CC 010A	FRAME	284TSC	SER No.	22005181167
HP	25	CLASS	F	HZ 60		
RPM	3520	PH	3	DES A		
VOLTS	230/460		CODE	J		
AMPS	58/29	USABLE:	AT 208 62 A			
RATING	AMB - CONT	NEMA NOM	EFF 91.7 %			
PF	88%	SERF	1.15	VPWM INVERTER READY CT 6 -60 Hz VT 3-60 Hz (20 I)		
USABLE AT	50 Hz. 20 HP	190/380V	56/28	1. OSF		

Sistema en funcionamiento normal

El sistema presenta una curva de corriente típica, con un pico de corriente de 42 A. en el arranque y posteriormente una señal estabilizada en 38A aproximadamente, corriente de trabajo, al evaluar estos datos el sistema lo reconoce como una codificación de fallo 0, lo que indica que trabaja de manera normal.

Figura 24

Interfaz del sistema en funcionamiento normal



Sistema con código de falla 1

El sistema presenta una curva de corriente con perturbaciones, con un pico de corriente de 46.4 A. y el valor medio de la corriente es de 42.5 A., valor que está por arriba de la corriente nominal, al evaluar estos datos el sistema lo reconoce como una codificación de fallo 1, lo que indica que existe un taponamiento en la salida.

Figura 25

Interfaz del sistema con codificación de fallo 1



Sistema con código de falla 2

El sistema presenta una curva de corriente con perturbaciones mayores que en la falla 1, no se observa un pico de corriente en el arranque y el valor medio de la corriente es de 31A, valor que está muy por debajo de la corriente de trabajo, al evaluar estos datos el sistema lo reconoce como una codificación de fallo 2, lo que indica que existe una saturación o taponamiento en la entrada.

Figura 26

Interfaz del sistema con codificación de fallo 2



Resumen de reportes de mantenimiento

Para validar el funcionamiento del sistema se realizaron pruebas controladas de las fallas más comunes que presenta el equipo, tomando en cuenta el tiempo que toma detectar una falla, el tiempo promedio de detección de fallas antes de la implementación del sistema es de 24 minutos, mientras que el tiempo promedio con el sistema implementado es de 8 minutos en promedio, lo que implica un tiempo de detección de falla, 3 veces menor al tiempo anterior.

Reportes de mantenimiento antes de la implementación del sistema

Tabla 2

Resumen de mantenimientos previa implementación del sistema

Núm. Reporte	Fecha	Hora inicio falla	Hora detección falla	Tiempo detección de falla	Falla	Observaciones
R0403	17-mar	7:28	7:49	0:21	Taponamiento de salidas	Falla de proceso
R0405	20-mar	15:34	15:53	0:19	Saturación en la entrada	Falla de proceso
R0406	21-mar	14:56	15:28	0:32	Sobreintensidad	Falla eléctrica
R0408	23-mar	9:23	9:52	0:29	Sobretensión en circuito intermedio	Falla eléctrica
R0410	23-mar	10:48	11:15	0:27	Subtensión en circuito intermedio	Falla eléctrica
R0410	25-mar	8:25	8:50	0:25	Pérdida de fase en la red	Falla eléctrica
R0412	28-mar	19:21	19:41	0:20	Pérdida de fase en el motor	Falla eléctrica

Reportes de mantenimiento con el sistema implementado

Tabla 3

Resumen de mantenimientos posterior a la implementación del sistema

Núm. Reporte	Fecha	Hora inicio falla	Hora detección falla	Tiempo detección de falla	Falla	Observaciones
R0403	03-abr	8:43	8:53	0:10	Taponamiento de salidas	Falla de proceso
R0405	05-abr	14:17	14:25	0:08	Saturación en la entrada	Falla de proceso
R0406	06-abr	9:52	10:02	0:10	Sobreintensidad	Falla eléctrica
R0408	08-abr	18:34	18:42	0:08	Sobretensión en circuito intermedio	Falla eléctrica
R0410	10-abr	11:23	11:30	0:07	Subtensión en circuito intermedio	Falla eléctrica
R0410	10-abr	6:48	6:57	0:09	Pérdida de fase en la red	Falla eléctrica
R0412	12-abr	17:04	17:12	0:08	Pérdida de fase en el motor	Falla eléctrica

Capítulo V

Conclusiones y recomendaciones

Conclusiones

- Se definió la corriente consumida por la bomba como parámetro principal para evaluar las fallas del sistema, debido a que es el valor que presentó una variabilidad más amplia en comparación a los demás parámetros tales como voltaje de salida, torque, temperatura y velocidad.
- Se implementó los protocolos de comunicación, Modbus TCP, Profinet, S7 y MQTT, para integrar los dispositivos del proceso y obtener de manera centralizada todos los datos del sistema.
- Se seleccionó el modelo matemático Medium Gaussian SVM, debido a que es el modelo más apropiado para sistemas de equipos rotativos, presentando el mejor desempeño en cuanto a confiabilidad y precisión.
- Se desarrolló el modelo matemático Medium Gaussian SVM para la identificación de fallas en una bomba centrífuga dentro de un sistema de ósmosis inversa, utilizando las herramientas *DiagnosticFeatureDesigner* y *Classification Learner*, para el desarrollo de algoritmos de mantenimiento predictivo dentro de Matlab.
- Se analizó las diferentes curvas de corriente, producidas durante una operación continua, los resultados presentaron variación dependiendo si se encontraba en condiciones normales o con alguna de las fallas tipificadas.
- Se diseñó una interfaz gráfica de usuario basada en una plataforma web para visualizar en tiempo real los parámetros del proceso, las curvas de las variables y los resultados obtenidos del sistema de predicción aplicado a los datos.
- El tiempo promedio de detección de fallas antes de la implementación del sistema es de 24 minutos, mientras que el tiempo promedio con el sistema implementado es de 8 minutos en promedio, lo que implica un tiempo de detección de falla 3 veces menor al tiempo anterior.

- Para próximas investigaciones, se puede generar un nuevo código de falla en bombas como puede ser falla por eje roto, también se puede generar falla por vibración en el motor como puede ser desalineamiento del eje del rotor, falla de rodamiento.

Recomendaciones

- Se recomienda adquirir un Broker MQTT privado, debido a que las versiones gratuitas de este servicio limitan el número de mensajes enviados, una opción viable es adquirir un servidor privado virtual (VPS), e instalar el bróker MQTT dentro de éste.
- Se recomienda que previo a la etapa de preprocesamiento se eliminen datos atípicos detectados por el sistema, esto permite una mejor discriminación de las curvas de interés dentro de la herramienta para el desarrollo de algoritmos de mantenimiento predictivo en Matlab, con la señal de tipo eléctrico que está dentro de la frecuencia de corte de 0.1 Hz se recomienda utilizar el tipo de filtro de promedio sincrónico de tiempo (TSA).
- La interfaz gráfica de usuario está alojada en un servidor VPS Hostinger que cuenta con un 1 gigabyte de memoria RAM, lo que abastece al procesamiento del sistema actual que evalúa los datos de un solo equipo, pero en caso un sistema de análisis de varios equipos, se recomienda adquirir un VPS con más espacio de procesamiento en RAM, para evitar que la página web donde se aloja la interfaz se ralentice.
- El sistema actual permite clasificar las fallas comunes presentadas por una bomba centrífuga, se recomienda para futuros trabajos de investigación añadir dispositivos que permitan obtener datos de vibración y de espectro térmico, parámetros utilizados para definir la vida útil restante (RUL) del equipo.
- Se recomienda crear una contraseña segura para el ingreso a la plataforma *NodeRed*, con la combinación de caracteres especiales, cifras, letras mayúsculas y minúsculas con el fin de evitar las vulnerabilidades y amenazas informáticas que son un riesgo para los sistemas y la información de la empresa, sobre todo en el entorno actual, altamente digitalizado y dependiente de los servicios TI.

Bibliografía

Alandí P. A., (2016) "Estudio de la implantación de Internet de las Cosas, en las redes Logísticas de la Cadena de Suministro".

Almada L. F., (2016). *The Industry 4.0 revolution and the future of Manufacturing Execution Systems (MES)*. *Journal of Innovation Management* 3: 16-21.

Antonopoulos, N., & Gillam, L. (2010). *Cloud Computing: Principles, Systems and Applications*.

Bidet-Mayer, T. (2016), *L'industrie du futur: une compétition mondiale*, Paris: Presses des Mines.

Brookshear J. G. (2012), *Computer Science. An Overview, 11th. ed. Addison-Wesley*, 2011. Traducido al español con el título Introducción a la computación. Pearson Educación, Madrid.

Đuričin, D. & Herceg, I. V. (2018). *Industry 4.0 and Paradigm Change in Economics and Business Management*. In: Ni, J. et al. (Eds.): AMP 2018, LNME, pp. 37–56.

García (2017), "Plataforma domótica basada en Raspberry Pi y el protocolo MQTT," Esc. Técnica Super. Ing. Informática y Telecomunicación.

García G. S., (2009), mantenimiento correctivo, organización y gestión de la reparación de averías, Editorial RENOVETEC.

Garlan, D., Perry, D. (1995) *Introduction to the Special Issue on Software Architecture, IEEE Transactions on Software Engineering*.

Garlan, D., Shaw, M. (1993). *An Introduction to Software Architecture. Advances in Software Engineering and Knowledge Engineering*.

Goti, A (2008) Sound-based predictive maintenance: a cost-effective approach, pp 37 – 40.

Goumagias, N., & Whalley, J., & Dilaver, O. & Cunningham, J. (2021), "Making sense of the internet of things: a critical review of internet of things.

IBM Watson and Bluemix. Acta Electrotécnica e Informática.

IBM, & Eurotech. (2010). *MQTT V3.1 Protocol Specification*, 1–42. Retrieved from

http://public.dhe.ibm.com/software/dw/webservices/wsmqtt/MQTT_V3.1_Protocol_Specific.pdf

Jamsa, C. (2012). *Cloud Computing*. (Jones & Bartlett Publishers 2012, Ed.).

Kezherashvili Beka (2018), Computación en la Nube, Universidad de Almería.

Kim, K.J. & Wang, S. (2021), "*Understanding the acceptance of the Internet of Things: an integrative theoretical approach*", *Aslib Journal of Information Management*.

Kumar D., Maurya R. K., & K. Dwivedi. (2019), "*IoT based home automation using Node-RED*," *Int. J. Innov. Technol. Explor. Eng.*, vol. 8, no. 12, pp. 5044–5047.

López F. M. (2019), Industria 4.0 para la monitorización de un proceso industrial.

Malhotra, N. K. (2004), Investigación de mercados matriz de confusión.

Mariscal Judith, & Gilgarcía Ramón. (2013). El cómputo en la nube en México: Alcances y Desafíos para los Sectores Público y Privado.

Martínez F. (2015). Gestión Integrada del Mantenimiento y la Energía para la Prevención de Fallos en Equipos de Plantas de Proceso. Universidad de Murcia.

Masip, R. O., Mantenimiento preventivo, Edición Departamento de Empresa y Empleo, pp 6-7.

Matsuo, K. & Barolli, L. (2020), "*IoT sensors management system using Agile-Kanban and its application for weather measurement and electric wheelchair management*", *International Journal of Web Information Systems*.

Oke, A.E. & Arowoia, V.A. (2021), "*Evaluation of internet of things (IoT) application areas for sustainable construction*", *Smart and Sustainable Built Environment*.

Olarte C., & Botero W. A., Cañon A., importancia del mantenimiento industrial dentro de los procesos de producción.

Orlowska T. & Wolkiewicz M., (2021), *Application of Machine Learning to a Medium Gaussian Support Vector Machine in the Diagnosis of Motor Bearing Faults, Electronics*.

Pacheco J., & Andrade J., & Rodríguez Yáñez S., (2018), *Cloud Computing para PYMEs*.

Ramírez Jair E. (2018), 1 Metodología para entrenamiento de máquina tipo jerárquico, para predecir el tipo de fallo y nivel de degradación en rodamientos, desde características de diferentes espacios de representación, Universidad Tecnológica de Pereira, Colombia.

Rincón M. N. (2021), Implantación de Mantenimiento 4.0 en una empresa del sector industrial, Universidad de Oviedo.

Sang, Go Muan, Lai Xu, Paul de Vrize, Yuewei Bai, Fangyu Pan, (2020), *PredictiveMaintenance in Industry 4.0, En Proceedings of the 10th Internacional Conference on Information Systems and Technologies, Lecce Italy: ACM*.

Sodring, T., & Reinholdtsen, P. & Massey, D. (2020), "*A record-keeping approach to managing IoT-data for government agencies*", *Records Management Journal*.

Sosinsky, B.(2011). *Cloud Computing Bible*. Wiley Publishing, Inc.. Indianapolis, Indiana.

Terán V. O., & Hernández G. P., & Flores L. J.(2017), Centro Universitario UAEM Amecameca; internet de las cosas (iot) como herramienta para la optimización de la cadena de suministro del sector secundario Revista Global de Negocios Vol. 5, No. 6, pp. 107-118.

Velásquez L. D., & Alba L., & Palencia A., & Suárez C.(2019), Aspectos Básicos de la Industria 4.0, Ministerio de Tecnologías de la Información y las Comunicaciones.

Anexos