Universidad Nacional de Loja

Facultad de la Energía, las Industrias y los Recursos Naturales no Renovables

Carrera de Ingeniería Electromecánica

“Uso de técnicas de Machine Learning para la detección de fallas en el módulo IGBT de los aerogeneradores del Parque Eólico Villonaco”

Trabajo de Titulación, previo a la obtención del título de Ingeniero Electromecánico

AUTOR:
Pablo José Caraguay Quinde

DIRECTOR:
Ing. Génesis Jahel Vásquez Rodríguez

Loja - Ecuador

2024

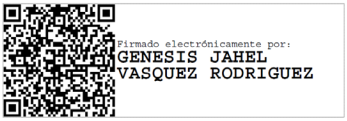
Certificación

Loja, 29 de abril del 2024.

Ing. Génesis Jahel Vásquez Rodríguez, Mg. Sc.
DIRECTORA DE TRABAJO DE INTEGRACIÓN CURRICULAR

CERTIFICO:

Que he revisado y orientado todo proceso de la elaboración del Trabajo de Integración Curricular denominado: "**Uso de técnicas de Machine Learning para la detección de fallas en el módulo IGBT de los aerogeneradores del Parque Eólico Villonaco**", previo a la obtención del título de **Ingeniero Electromecánico** de la autoría del estudiante **Pablo José Caraguay Quinde**, con cédula de identidad Nro. **1105656134**, una vez que el trabajo cumple con todos los requisitos exigidos por la Universidad Nacional de Loja, para el efecto, autorizo la presentación del mismo para su respectiva sustentación y defensa.



Ing. Génesis Jahel Vásquez Rodríguez Mg. Sc**.**
DIRECTORA DEL TRABAJO DE INTEGRACIÓN CURRICULAR

Autoría

Yo, **Pablo José Caraguay Quinde**, declaro ser el autor del presente Trabajo de Integración Curricular y eximo expresamente a la Universidad Nacional de Loja y a sus representantes jurídicos de posibles reclamos o acciones legales, por el contenido del mismo. Adicionalmente acepto y autorizo a la Universidad Nacional de Loja, la publicación de mí Trabajo de Integración Curricular en el Repositorio Institucional – Biblioteca Virtual.

Firma: 

Fecha: 29 de abril del 2024

Cédula: 1105656134

Teléfono: 0992204674

Correo electrónico: pablo.caraguay@unl.edu.ec

Carta de autorización por parte del autor, para consulta, reproducción parcial o total y/o publicación electrónica del texto completo, del Trabajo de Integración Curricular.

Yo, **Pablo José Caraguay Quinde,** declaro ser autor del trabajo de integración curricular denominado: "**Uso de técnicas de Machine Learning para la detección de fallas en el módulo IGBT de los aerogeneradores del Parque Eólico Villonaco**". como requisito para optar el título de **Ingeniero Electromecánico,** autorizo al Sistema Bibliotecario de la Universidad Nacional de Loja para que con fines académicos, muestre al mundo la producción intelectual de la Universidad, a través de la visibilidad de su contenido de la siguiente manera en el Repositorio Institucional.

Los usuarios pueden consultar el contenido de este trabajo en el Repositorio Institucional, en las redes de información del país y del exterior, con las cuales tenga convenio la Universidad.

La Universidad Nacional de Loja, no se responsabiliza por el plagio o copia del trabajo de titulación que realice un tercero.

Para constancia de esta autorización, suscribo, en la ciudad de Loja, a los veintinueve días del mes de febrero del dos mil veinticuatro.

Firma:

Autor: Pablo José Caraguay Quinde.

Cédula: 1105656134.

Dirección: Av. Oriental de paso y Génova

Teléfono: 0992204674.

Correo electrónico: pablo.caraguay@unl.edu.ec

DATOS COMPLEMENTARIOS:

Directora del Trabajo de Integración Curricular: Ing. Génesis Jahel Vásquez Rodríguez Mg. Sc.

Dedicatoria

Dedico este trabajo de titulación a mis padres, José Benigno Caraguay Condor y María del Cisne Quinde Sánchez por el esfuerzo tan grande que han llevado a cabo a lo largo de toda mi vida al proveerme de lo necesario para mi formación intelectual y emocional.

A mis hermanos por darme su apoyo y un motivo por el cual esforzarme. A toda mi familia en general por la atención que me prestaron en cada aspecto de mi vida.

A mis compañeros porque entre todos nosotros hemos batallado para llegar hasta este punto y, finalmente, a todas aquellas personas que me han brindado sus palabras de aliento o han dedicado un momento de su tiempo a escuchar mis problemas y prestarme su ayuda.

Pablo José Caraguay Quinde

Agradecimientos

Agradezco a mis padres por darme su apoyo durante mi formación académica y el afecto tan grande brindado a lo largo de mi vida.

A mis hermanos por la convivencia diaria, su apoyo en momentos de crisis y tantos momentos divertidos que hemos vivido.

A los diferentes docentes que han compartido sus conocimientos con mi persona a lo largo de la carrera, especialmente al Ing. Jorge Maldonado por brindarme las pautas necesarias para iniciar con este proyecto y a la Ing. Génesis Vásquez por guiarme a través del mismo.

Pablo José Caraguay Quinde

Índice de contenidos

Índice de tablas:

Índice de figuras:

Índice de anexos:

Acrónimos:

AUC: Area under the curve.

CELEC: Corporación eléctrica del Ecuador.

CEV: Central eólica Villonaco.

CVS: Archivo que permite guardar datos en formato de tabla estructurada.

FN: Falsos Negativos.

FP: Falsos positivos.

GB: GigaBytes

IGBT: Insulated gate bipolar transistor.

Kg: kilogramos.

KNN: K-nearest neighbors.

m/s: metros sobre segundos.

ML: Machine Learning

MP: Mantenimiento preventivo.

MSNM: metros sobre el nivel del mar.

MW: Megavatios.

NB: Naive Bayes.

RF: Random Forest.

ROC: Receiver Operator characteristic.

SCADA: Supervisory Control and Data Acquisition.

SPSS: Statistical Package for Social Sciences.

SVM: Máquinas de vectores de soporte.

TPM: Mantenimiento productivo total.

VN: Verdaderos negativos.

VP: Verdaderos Positivos.

1. Título

“Uso de técnicas de Machine Learning para la detección de fallas en el módulo IGBT de los aerogeneradores del Parque Eólico Villonaco”.

2. Resumen

Ante la necesidad de reducir el consumo de fuentes no renovables al momento de producir energía eléctrica, se han planteado varias alternativas entre las que destaca la energía eólica. Este tipo de energía limpia posee una serie de desventajas que impiden su completo desarrollo e implementación. La falla de sus componentes es una de las más frecuentes y costosas de tratar debido a la incapacidad que se tiene para detectarla e impedirla.

Para el presente trabajo se cuenta con la base de datos de los valores captados por los sensores instalados en los diferentes componentes de los aerogeneradores y recogida por el sistema SCADA de la central eólica Villonaco. Esta cuenta con 70 variables de las cuales 22 serán usadas en el entrenamiento de algoritmos de Machine Learning que tienen como objetivo la detección de los momentos en que la alarma correspondiente a la falla en el módulo IGBT del convertidor de potencia de los aerogeneradores instalados en dicha central se enciende.

Dado que se desconoce la calidad de la base de datos, esta es sometida a un preprocesamiento en el que se hace la limpieza de valores que afecten negativamente a los algoritmos a usar. Estos pueden ser valores nulos, repetidos, irrelevantes, etc. Esto se hace por medio del estudio de las características de dicha base por medio del software SPSS y su posterior tratamiento con Python.

Los algoritmos Random Forest, KNN, SVM y NB son entrenados con ayuda del software Python y las herramientas que posee el entorno de Google Colab para el trabajo de Machine Learning. Dichos algoritmos son evaluados por métricas de rendimiento que muestran el comportamiento de los algoritmos y el desempeño al momento de predecir las fallas.

Finalmente, para una mejor comprensión de los resultados, se usa el método gráfico denominado curvas ROC, el cual es una métrica que permite comparar visualmente la eficiencia de cada uno de los algoritmos.

Todos estos pasos nos llevan a la conclusión que el algoritmo RF fue quien mejor desempeño tuvo al tener un valor de precisión de 88,25%, seguido de KNN, SVM y Naive Bayes con valores de 85%, 88,25% y 73.1% respectivamente. Dado que todos los valores de precisión se encuentran por arriba del 0.5% se puede concluir que todos han logrado detectar el momento en que la alarma de falla del módulo IGBT se acciona, pero con diferentes desempeños.

Palabras claves: Machine Learning, SCADA, IGBT.

Abstract

In order to reduce the consumption of non-renewable sources when electrical energy is producing, several alternatives have been proposed, among which wind energy stands out. This type of clean energy has a series of disadvantages that hinder its complete development and implementation. Component failure is one of the most frequent and costly to deal with due to the inability to detect and prevent it.

For this academic study, we have the database of values captured by sensors installed in the different components of the wind turbines and collected by the SCADA system of the Villonaco wind farm. This database consists of 70 variables, of which 22 will be used in the training of Machine Learning algorithms aimed at detecting the time when the alarm corresponding to the failure in the IGBT module of the power converter of the turbines installed in this electric wind farm.

Since the quality of the database is unknown, it undergoes preprocessing to clean values that negatively affect the algorithms to be used. These can be cero values, duplicates, irrelevant data, etc. This is done through the study of the characteristics of the database using SPSS software and its subsequent processing with Python.

The Random Forest, KNN, SVM, and NB algorithms are trained using Python software and the tools available in the Google Colab environment for Machine Learning work. These algorithms are evaluated by performance metrics that show the behavior of the algorithms and their performance when predicting failures.

Finally, for a better understanding of the results, the graphical method called ROC curves is used, which is a metric that allows visually comparing the efficiency each one of the algorithms.

All these steps lead us to the conclusion that the RF algorithm had the best performance with a precision value of 88.25%, followed by KNN, SVM, and Naive Bayes with values of 85%, 88.25%, and 73.1%, respectively. Since all precision values are above 0.5%, it can be concluded that they have all managed to detect the exactly time when the IGBT module failure alarm is triggered, but with different performances.

Keywords: Machine Learning, SCADA, IGBT, Random Forest.

3. Introducción

Las energías renovables han tomado fuerza en los últimos años en todo el mundo debido a la necesidad energética que tiene el mundo actualmente. Esto, sumado a la creciente preocupación mundial debido a la contaminación causada por el uso de combustibles fósiles, ha llevado a que se desarrollen nuevas formas de aprovechar los recursos naturales renovables, lo que ha convertido a este campo como uno de los que más investigación requiere.

Uno de los más conocidos y prometedores métodos de producción energética es la energía eólica, considerada una de las más importantes y con la que se han llevado a cabo un sinnúmero de investigaciones. Las características de la energía eólica, han sido de tal importancia que, alrededor del mundo se instalen centrales energéticas que aprovechan la fuerza del viento para generar energía sin la necesidad de combustibles contaminantes.

De acuerdo con Maldonado (2015), la velocidad promedio del viento en la ciudad de Loja, en la estación de Argelia, es de aproximadamente 4 m/s. De acuerdo a la página de la fabricadora de aerogeneradores Goldwind, la velocidad mínima para el funcionamiento de los mismos es de 2.5 m/s, lo cual demuestra que la ciudad de Loja posee en ambiente favorable para la generación eólica. Dicho esto, se tiene que en Ecuador se han establecido tres parques eólicos de los cuales, la Central Eólica Villonaco es la más llamativa debido a la importancia que tiene a nivel nacional, así como sus características de construcción y producción.

La central Villonaco cuenta con 11 aerogeneradores tipo GoldWind GW70/1500 para una potencia instalada total de 16.5 MW. Esta central es la referente ecuatoriana en cuanto energía eólica se refiere puesto que cubre aproximadamente el 30% de la demanda energética de las provincias de Loja, Zamora y parte de Morona Santiago.

La central eólica Villonaco cuenta con un factor de planta del 41%; es decir, produce el 41% de lo que generaría si se trabajara a plena potencia durante todo un año. Es casi imposible conseguir un factor de potencia del 100% puesto que esto solo ocurre en la teoría, pero con un mantenimiento correcto se podría elevar ese factor de potencia.

Las fallas mecánicas en la central eólica Villonaco provocan la parada de los aerogeneradores y, por ende, el cese de producción energética durante el tiempo que se tarden las reparaciones necesarias para trabajar con seguridad. Este tipo de fallas pueden surgir de diversos componentes, pero existe uno que ocasiona pérdidas considerables cada vez que falla: el módulo IGBT.

De acuerdo a Canteli (2013), el módulo IGBT es un dispositivo destinado a la conversión de potencia y transmisión de energía de los aerogeneradores. Es un dispositivo de

tres terminales con capacidad de control externo que posee una baja resistencia a la corriente y una elevada velocidad de conmutación.

El módulo IGBT está ubicado en el convertidor del aerogenerador y su importancia radica en que el fallo de estos implica su destrucción por lo que es necesario reemplazarlo por uno nuevo. Esto genera grandes pérdidas puesto que, además de las generadas por el paro del aerogenerador se debe incluir el costo del módulo IGBT que debe ser exportado desde la fábrica ubicada en el extranjero. Se desconoce la cantidad exacta que implica todo este procedimiento debido a que estos datos son confidenciales, pero se sabe que es una cantidad fuerte.

Las investigaciones sobre mantenimiento han llevado a la conclusión de que el mantenimiento predictivo es el más adecuado para aerogeneradores. Una de las conclusiones del mantenimiento predictivo dicta que "partiendo de datos del sistema SCADA y de equipos específicos de monitorización, se pueden construir modelos globales del aerogenerador que también permiten predecir algunas variables de estado en función de otras, de manera que se pueda comparar la predicción con la medición" (*Romero Lozano, 2016)*. La conveniencia de esta conclusión recae en que la central eólica Villonaco cuenta con un sistema SCADA que reúne datos de los sensores instalados cada 10 minutos, las 24 horas del día durante los 365 días del año.

En este proyecto, se utilizarán técnicas de análisis de datos y de aprendizaje automático para analizar los datos del sistema SCADA. El objetivo es limpiar los datos del sistema SCADA, seleccionar las variables que guardan relación con el fallo del módulo y aplicar algoritmos de aprendizaje automático supervisado para clasificar cuando se produce una falla en el IGBT.

Antes de aplicar los modelos de clasificación, los datos deben pasar por una etapa de preprocesamiento, esta etapa sirve para eliminar datos que podrían interferir en el análisis posterior.

Para finalizar. se validarán los resultados de los modelos de clasificación aplicados a través de métricas que permitirán evaluar el desempeño de los algoritmos.

Objetivos

Objetivo General

Aplicar herramientas de Data Mining y Machine Learning para clasificación y detección de fallas en los módulos IGBT de los aerogeneradores del parque eólico Villonaco.

Objetivos específicos

- Usar técnicas de Data Mining para el preprocesamiento de los datos obtenidos por el sistema SCADA del parque eólico Villonaco.
- Aplicar los algoritmos de clasificación: KNN (Vecinos Cercanos), SVM (Máquinas de Soporte), RF (Random Forest) y NB (Clasificador Bayesiano) para la clasificación/detección de fallas.
- Evaluar y validar los algoritmos por medio de métricas de rendimiento para determinar el algoritmo con el mejor desempeño.

4. Marco teórico

4.1 Capítulo I: Estado del arte y fundamentos teóricos.

4.1.1 Energía Eólica

La energía eólica se caracteriza por ser una energía ilimitada y no contaminante que está teniendo un fuerte impacto a nivel mundial, tanto por sus implicaciones económicas como sociales. Esta energía, que representa el 1% o 2% de la energía solar que ingresa al planeta, se genera debido a que las diferencias de temperatura promueven la circulación del aire de la atmósfera. Las zonas cercanas al Ecuador reciben mayor energía solar por lo que también poseen un mayor potencial eólico (Suarez & Carbajal, 2008).

La energía eólica ha sido aprovechada de varias formas a lo largo de la historia de la humanidad: los antiguos egipcios usaron naves con velas para trasladarse por el río Nilo hace 5000 años y, más recientemente, en 1979 se construyeron barcos que usan energía eólica para reducir el consumo de combustible (Narvaez, 2021) pero si nos referimos a generación energética, es necesario hablar de los aerogeneradores. Estos instrumentos aprovechan la energía del viento para hacer girar unas aspas y, por medio de ellas, transformar la energía mecánica a eléctrica por medio de un generador.

4.1.1.1 Origen de los vientos.

De acuerdo a Narvaez "el calentamiento dispar de la superficie terrestre por acción de la radiación solar es el principal causante de los vientos" (p.2). Esto se debe a que la radiación solar es absorbida por la tierra, calienta el aire y este se eleva en los trópicos de modo de desplazar al aire frío proveniente de los polos terrestres. Como la radiación varía acorde los movimientos del planeta se tienen estaciones con más vientos y estaciones con menos vientos.

4.1.2 Central Eólica Villonaco.

La central eólica Villonaco, de 16.5 MW de potencia fue empezada a construir durante el mes de agosto del año 2011 en la ciudad de Loja, al sur del Ecuador. Cuenta con 11 aerogeneradores GoldWind GW70/150 de 1.5 MW de potencia cada uno (Soto, 2017).

Esta central cuenta con una velocidad de viento estimada de más de 12 m/s al año, con una densidad de aire de 0.923 kg/m^3 a una altura de 2700 msnm

4.1.3 Aerogeneradores

Los aerogeneradores son estructuras capaces de transformar corriente mecánica en corriente eléctrica por medio del movimiento de unas alas unidas a un eje. Las alas son puestas en movimiento por la fuerza de las corrientes de aire sobre la superficie de estas.

Los aerogeneradores pueden ser divididos en tres clases que van a depender de la posición del aerogenerador, la posición del equipo respecto al viento y por el número de palas (Villarubia López, 2011).

- Por la posición del aerogenerador:
  - Eje vertical
  - Eje Horizontal
- Por la posición del equipo con respecto al viento:
  - A barlovento
  - A sotavento
- Por el número de palas:
  - Una pala
  - Dos palas.
  - Tres palas.
  - Múltiples.

4.1.3.1 Aerogeneradores de eje vertical.

Este tipo de aerogeneradores tienen poca demanda en comparación con los de eje horizontal por razones técnicas y económicas. Según Villarubia, estos aprovechan se dividen dos:

- De arrastre diferencial (rotor Savonius):
  Usa la diferencia de la fuerza que el viento genera entres dos superficies, una convexa y otra cóncava. Está conformado por dos semicilindros con los ejes separados. Esta separación ayuda a que el viento fluya por lo que también se aprovecha esta variación de cantidad de movimiento (ver Figura 1).
- De variación cilíndrica (rotor Darrieus):
  Este rotor está formado por un conjunto de álabes arqueados biconvexos con superficies esféricas, parabólicas o cilíndricas que giran alrededor de un eje vertical que posee un perfil aerodinámico.

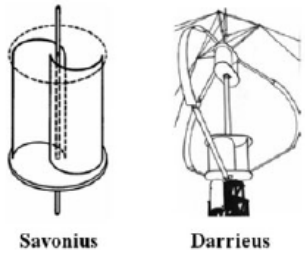


***Figura** 1: Aerogenerador de arrastre diferencial*

Fuente: (Miguel Villarubia López, 2012)

4.1.3.2 Aerogeneradores de eje horizontal.

Los aerogeneradores de eje de tipo horizontal pueden ser clasificados de acuerdo a su velocidad de giro (ver **Tabla** 1):

- Generadores lentos.
- Generadores rápidos.
- Generadores de velocidad intermedia.

Tabla 1**:** Tipos de aerogeneradores de acuerdo a su velocidad.

Tipos de generadores	**Características**	**Velocidades de viento de arranque**	**Usos**
Generadores lentos	Multipala. Baja velocidad específica de diseño. Gran par de arranque.	Entre 2 m/s y 3 m/s	Accionamiento de bombas de agua.
Generadores rápidos	Dos o tres palas. Velocidad específica alta. Par de arranque bajo.	Entre 3 m/s y 4 m/s.	Generación eléctrica en parque eólicos.
Generadores de velocidad intermedia	Prestaciones comprendidas entre las dos anteriores.	Velocidades comprendidas entre las dos anteriores.	Equipos autónomos para producción de electricidad.

Fuente: (*Villarubia López*, 2012)

Generalmente los aerogeneradores de eje horizontal cuentan con los siguientes elementos:

- Base.
- Punto de conexión a la estación.
- Escalera interior.
- Sistema de orientación.
- Anemómetro y veleta.

- Rotor.
- Caja de engranajes.
- Alternador o generador eléctrico.
- Góndola en cuyo interior se encuentran mecanismos tales como el multiplicador de velocidad, sistemas auxiliares de regulación y control.
- Torre de sustentación de todo el conjunto

Estos aerogeneradores son de vital importancia en esta tesis puesto que a esta clase pertenecen los aerogeneradores GoldWind GW70/1500, que son los instalados dentro de la central eólica motivo de estudio.

4.1.4 Convertidores de potencia.

Un convertidor de energía es un sistema o dispositivo electrónico cuyo propósito es convertir energía eléctrica entre dos formatos diferentes.

4.1.4.1 Tipos de convertidores de potencia

Los convertidores se pueden clasificar según diferentes criterios, pero generalmente se agrupan por la energía de entrada y salida. Por medio de este estándar se encuentran crear cuatro grupos.

4.1.4.1.1 Convertidores ca/cc o rectificadores

Transforman corriente alterna monofásica o trifásica en corriente continua (Capuma Condori, 2018).

4.1.4.1.2 Convertidores cc/cc.

Transforman cierta cantidad de corriente continua de entrada en una de salida (Capuma Condori, 2018).

4.1.4.1.3 Convertidores cc/ca

También llamados inversores, transforman corriente continua en corriente alterna con la opción de controlar frecuencias, valores eficaces, etc. Suelen ir asociados a un rectificador (Capuma Condori, 2018).

4.1.4.1.4 Convertidores ca/ca.

Generalmente usados en el diseño de arrancadores suaves porque pueden modificar el valor eficaz de la tensión de entrada sin alterar la frecuencia (Condori, 2018).

4.1.4.2 Semiconductores de potencia.

El primer material usado en la construcción de un convertidor fue el mercurio en el año de 1900. En la década de los 50 empezó un desarrollo de nuevos materiales luego de que se inventara el transistor de silicio en los laboratorios Bell. Actualmente existen una gran cantidad

de tecnologías para la fabricación de convertidores con la semejanza de que todas persiguen el mismo resultado (Canteli, 2013).

4.1.4.3 Diodo.

Semiconductor que une dos terminales conocidos como ánodo y cátodo. Este es el elemento más usado a pesar de que es un elemento no controlado debido a que su conmutación del estado de bloqueo al de conducción o viceversa depende del signo de la intensidad que lo recorre (Rubio Peña, 2012).

4.1.4.4 Tiristor.

Los tiristores son fabricados por medio de la unión de 4 capas con una estructura de tres uniones pn. Al igual que los diodos posee ánodo y cátodo, pero además cuenta con una puerta que tiene la función de recibir corriente para que el tiristor empiece a funcionar como diodo (*Aguilar*, 2020).

4.1.4.5 MOSFET.

El MOSFET es un dispositivo que permite una rápida velocidad de conmutación por medio de una pequeña intensidad de entrada. Dispone de tres terminales denominados drenador, puerta y surtidor. El surtidor y drenador son los polos del interruptor equivalente y la puerta como elemento de control (Canteli, 2013).

4.1.4.6 Transistor IGBT.

El Insulated gate bipolar transistor o IGBT es un dispositivo de tres terminales con capacidad de control externo. Fue desarrollado para cubrir la necesidad de aprovechar la ventaja de baja resistencia que poseen otros transistores y la elevada velocidad de conmutación de los MOSFET. Este dispositivo se controla por medio de tensión además de poseer un coeficiente de temperatura positivo que permite su funcionamiento en paralelo y si estos módulos reciben una sobrecarga van a aumentar su resistencia de conducción y reducir su carga. (Condori, 2018).

Dadas las características de tensión e intensidad medias, caídas de tensión en conducción, frecuencia de conmutación y controles de puerta, los módulos IGBT pueden ser considerados como una especie de híbridos entre los transistores bipolares BJT y MOSFET. Lo anteriormente dicho se valida al saber que los transistores IGBT combinan la capacidad de los transistores BJT para soportar corrientes altas y la capacidad de controlar mediante un driver o dispositivo de control la conducción del dispositivo por tensión de los MOSFET, aunque las pérdidas en conmutación de los IGBT son mayores.

La configuración de los transistores IGBT es de cuatro capas: N^+, P, N^-, P^+. En caso de que se incluya una capa adicional, es decir un búfer, la configuración queda de 5 capas N^+, P,

N^-, N^+, P^+. Según Fernández, (2021) “Esta estructura aumenta la densidad de corriente y disminuye la tensión en conducción” (p. 5).

4.1.4.6.1 Modos de operación del IGBT.

Conducción directa: Por medio de la aplicación de tensión positiva entre las terminales la corriente va a pasar a través del IGBT de colector a emisor. Durante este momento existe una caída de tensión entre colector y emisor debido a la resistencia del IGBT.

Este modo de operación es interesante debido a que permite tener corriente entre colector y emisor controlada mediante la tensión de puerta con una caída de tensión de puerta, con una caída muy baja.

Apagado o bloqueo: Se corta el paso de la corriente con una tensión nula por lo que el dispositivo empieza a soportar una tensión positiva entre colector y emisor (tensión de bloqueo).

Conducción inversa: El diodo colocado en antiparalelo comienza a conducir en sentido contrario al IGBT. Esto se intenta evitar puesto que la corriente aumenta de forma descontrolada.

4.1.4.7 Módulo IGBT

La versatilidad de esta clase de transistores se debe a la integración de los mismos en módulos. Esta disposición hace más fácil el diseño (**Figura** 2), pero implica la consideración de otros factores eléctricos y térmicos que se dan cuando este dispositivo está en funcionamiento (Lorenzi, 2002).

El módulo IGBT consiste en dos transistores IGBT´s con sus diodos en antiparalelo. De acuerdo a Fernández (2021), estos módulos son capaces de conmutar a frecuencias de hasta 12kHz.

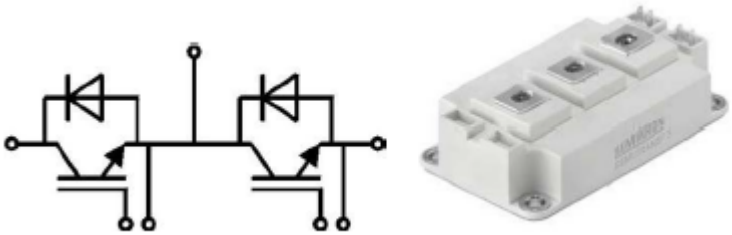


Figura 2**:** *Esquema / imagen de un módulo IGBT.*

Fuente:(Fernández, 2021)

El principio de funcionamiento básico de un IGBT implica el encendido del MOSFET mediante un voltaje positivo desde la puerta hasta el emisor, permitiendo que el voltaje conectado al colector conduzca la corriente de base a través del transistor bipolar y el MOSFET.

Esto enciende el transistor bipolar, permitiendo el flujo de corriente de carga. Por otro lado, un voltaje negativo en la puerta apaga el MOSFET, interrumpiendo la corriente de base y apagando el transistor bipolar.

A pesar de su simplicidad conceptual, el desarrollo de hardware para controlar un IGBT puede resultar complejo debido a las variaciones de rendimiento en los dispositivos y circuitos reales. En la mayoría de los casos, los fabricantes de semiconductores ofrecen soluciones integradas, como controladores de puerta, con una amplia variedad de funciones y capacidades, lo que simplifica enormemente esta tarea. Es fundamental asegurar la compatibilidad entre los módulos IGBT y los controladores de puerta adecuados (Fernández, 2021).

Los módulos IGBT están disponibles en una amplia gama de paquetes, desde tamaños más grandes con una capacidad nominal de 3,300 voltios o más, diseñados para aplicaciones de megavatios como sistemas de energía renovable y motores grandes, hasta tamaños medianos con tensiones nominales de 600 a 1700 voltios, utilizados en aplicaciones como vehículos eléctricos, motores industriales e inversores solares.

4.1.4.8 Fallas en el módulo IGBT

De acuerdo a Narváez (2019):

> El volumen de negocio que genera la industria Eolo eléctrica es sobresaliente, principalmente para los fabricantes de aerogeneradores quienes reciben alrededor del 75% del costo de inversión de una central Eolo eléctrica. Algunos fabricantes han estado tan ocupados produciendo y vendiendo máquinas que, hasta cierto punto, han postergado invertir en desarrollos de nueva generación o al menos en dar soluciones plausibles y definitivas a una serie de problemas técnicos recurrentes que se han hecho evidentes con la aplicación masiva de la tecnología. (p. 55).

Esto indica que muchos de los problemas que se generan en los generadores no reciben la atención que se merecen dado que los fabricantes no perciben una compensación económica sustancial por lo que la tecnología eólica no ha alcanzado su verdadera madurez hasta el día de hoy y esto se puede ver claramente

Existen ocasiones en las que el módulo IGBT no puede disipar correctamente el calor por lo que su temperatura interna sube a niveles capaces de generar gas que termina por romper la carcasa. A este fenómeno se lo conoce como explosión IGBT

De acuerdo a Ghassemi (2018), las explosiones de los módulos IGBT se pueden dar por diferentes motivos:

- Potencia de calentamiento excede a la potencia de disipación de calor por lo que lo que se provoca un sobrecalentamiento.
- Errores al momento de la instalación: línea entrante conectada al terminal de la línea que sale, mala conexión a la fuente de alimentación o a la carga.
- Sobrecorrientes que alteran la lógica en el circuito de control.
- Daño del aislamiento.
- Sobrevoltaje ocasionado por el pico de voltaje generado por la inductancia parásita de la línea.
- El circuito de detección de flujo se encuentra desfasado con el del tiempo de reacción.
- Problemas de proceso tales como una fila de cobre fuerte o falta de tensión en los tornillos.
- Etc.

Existen muchas otras posibilidades para que un módulo IGBT explote, pero otros autores mencionan que las fallas en los módulos IGBT incluyen el fallo al iniciar su funcionamiento, despegue del cable de unión de cortocircuito o fatiga en las soldaduras. Se dice que se puede realizar un monitoreo de la degradación de estos módulos al usar variables como el voltaje de saturación del colector emisor, voltaje de umbral del emisor de puerta, resistencia reactiva y resistencia interna (Liton, 2018).

4.1.5 Mantenimiento.

El mantenimiento es el resultado de la necesidad de asegurar un correcto desempeño de los equipos y conseguir la disponibilidad más elevada.

El concepto de mantenimiento varía de acuerdo al autor y al tema en cuestión, pero la gran mayoría coincide en que se trata de "mantener el correcto estado funcional de los equipos e instalaciones" (Gómez, 1998).

El estudio del mantenimiento se ha dado en las últimas décadas y, a pesar del poco tiempo invertido, se han conseguido desarrollar metodologías, conceptos y niveles de aplicación. Es de esta forma que se ha conseguido pasar de simples reparaciones de mecanismos y equipos a un complejo sistema de gestión optimizada de recursos técnicos capaces de prevenir, corregir y/o predecir averías en determinados elementos.

A pesar de todo lo dicho anteriormente, se debe tener en cuenta que la carencia de materias instructivas en los planes de estudio enfocados en tecnología y la falta de interés en estos temas han provocado un déficit informativo y formativo para los profesionales de este campo.

4.1.5.1 Tipos de mantenimiento

Acorde a las posibles funciones que se tomen en consideración, el mantenimiento puede ser clasificado en los siguientes tipos (*Cesáreo*, 1998):

4.1.5.1.1 Mantenimiento Correctivo.

Este tipo de mantenimiento se da en equipos cuando ya ha sucedido la falla. Se trata de una actitud no activa puesto que simplemente se espera a que la falla se dé. Esta clase de mantenimiento se practica en una gran cantidad de industrias, aunque, en la gran mayoría de casos, se aplica únicamente en elementos cuya falla no implique la interrupción de la producción (*Gómez*, 1998).

4.1.5.1.2 Mantenimiento productivo total.

El mantenimiento productivo total o TPM por sus siglas en inglés, recoge todos los aspectos que tienen que ver con el uso de los equipos e instalaciones y asigna tareas que se desglosan hasta llegar a los operadores que llevarán a cabo estas acciones específicas sobra cada componente.

4.1.5.1.3 Mantenimiento preventivo.

Este tipo de mantenimiento tiene el objetivo de disminuir o evitar la reparación total de un componente mediante una serie de inspecciones periódicas en las que renuevan partes deterioradas, sustituyen componentes menores, etc. El correcto funcionamiento de esta clase de mantenimiento depende de la cantidad de tiempo con la que se realicen las inspecciones.

De acuerdo con Lozano (2016), el mantenimiento preventivo surgió debido a la necesidad de mejorar los resultados obtenidos por medio del mantenimiento correctivo puesto que “con el preventivo se abarca el estudio de la documentación, medidas de parámetros y procedimientos de obtención de los mismos” (*Lozano*, 2016).

4.1.5.1.1 Mantenimiento predictivo.

La idea de este tipo de mantenimiento es la de reemplazar inmediatamente los componentes cuando no se encuentren en buenas condiciones operativas. De este modo se evitan las inspecciones periódicas. Para realizar este trabajo se debe de poseer datos con antelación que indiquen el estado del equipo y mantener vigilado continuamente la maquinaria.

Este tipo de mantenimiento aplica nuevas tecnologías para “determinar la situación del equipo, monitorear su condición a lo largo del tiempo y lograr la detección de fallas e

intervenciones en el momento justo en que deben realizarse por medio de la corrección de la raíz del problema" (Lozano, 2016).

Romero Lozano Luis dice que el mantenimiento predictivo es el más adecuado para aerogeneradores debido a que permite predecir con suficiente antelación el momento en que la falla va a suceder y de esta forma se pueden tomar medidas para evitar que esto pase. Esto implica periodos cortos de parada y disminución de las pérdidas monetarias en la central eólica.

Romero Lozano Luis también nos indica las conclusiones acerca del mantenimiento predictivo:

a) La variedad de solicitaciones que soportan los aerogeneradores requiere implantar técnicas de mantenimiento predictivo.
b) Para que un tipo de averías sea gestionable con técnicas de MP se precisa conocer sus síntomas y su capacidad para detectar los niveles de alerta con suficiente antelación para programar intervenciones.
c) El MP se puede abordar con inspecciones del estado real de los componentes, con monitorización del comportamiento de los componentes o midiendo y calculado las solicitaciones por deterioro que experimentan los componentes que las soportan.
d) Principales técnicas de las condiciones monitorizadas que se aplican actualmente en eólicas son: análisis de temperaturas, análisis de vibraciones, contaminación por partículas en aceite y deformaciones en palas.
e) Partiendo de datos del sistema SCADA y de equipos específicos de monitorización, se pueden construir modelos globales del aerogenerador que también permiten predecir algunas variables de estado en función de otras, de manera que se pueda comparar la predicción con la medición.
f) Los contadores de fatiga en su formulación requieren de un tratamiento avanzado de ingeniería, pero una vez establecidos permiten de una manera sencilla obtener indicadores del deterioro de componentes en base a medidas sencillas como la velocidad media de viento (*ROMERO*, 2016).

Para este trabajo se tomará en cuenta la conclusión "*e"* puesto que usando los datos del sistema SCADA se modelarán algoritmos de Machine Learning para la predicción de fallas en el módulo IGBT.

4.1.6 Sistema SCADA.

El sistema SCADA (Supervisory Control And Data Adquisition) es un sistema que permite el acceso a datos remotos de uno o varios determinados procesos al mismo tiempo que nos da control sobre estos mismos (*Rodríguez, 2012*).

4.1.6.1 Arquitectura general de un sistema SCADA.

El sistema queda dividido en tres bloques principales (*Rodríguez*, 2012).

- Software de adquisición de datos y control (SCADA)
- Sistemas de adquisición de mando (sensores y actuadores).
- Sistema de interconexión (comunicaciones.

Mediante el software de adquisición de datos es que el sistema SCADA recoge los datos que permiten la creación de estrategias dentro de las empresas

La arquitectura del sistema SCADA se muestra en la **Figura** 3.

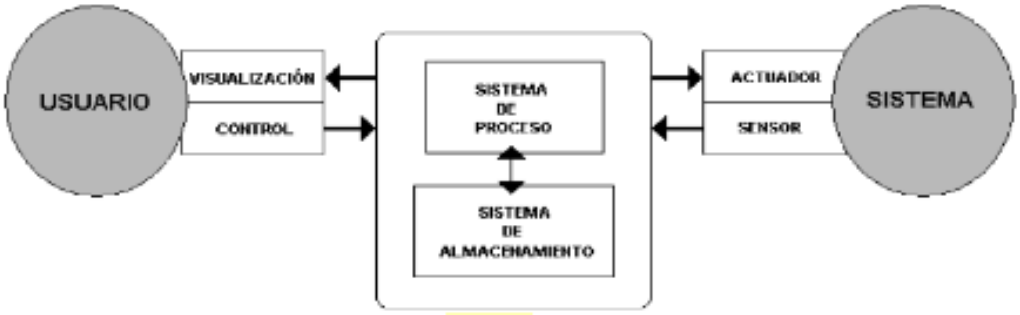


Figura 3**:** *Arquitectura de un sistema SCADA*

Fuente: **(*Rodríguez,* 2012)**

En la automatización es de gran utilidad disponer de una base de datos en la que consten las variables del sistema y, por medio de estas, se pueda trabajar con diferentes estrategias para realizar una diversa cantidad de análisis.

4.1.6.2 Alarmas y eventos

Se basan en la vigilancia de los parámetros del sistema puesto que están referidos a los acontecimientos no deseados puesto que son dañinos y perjudican el funcionamiento. Estas alarmas requieren de la intervención del operario para dar una solución adecuada al problema.

Los eventos son el resto de situaciones donde el funcionamiento es normal. Estos no requieren de la intervención del operario debido a su naturaleza benigna.

4.2 Capítulo II: Data Mining y aprendizaje automático

4.2.1 Data Mining

Se conoce como Data Mining al conjunto de técnicas utilizadas para interpretar la información de grandes volúmenes de datos. Estas técnicas buscan darle sentido a la

información almacenada puesto que el acceso a estos datos, en un futuro, causará un gran impacto en la sociedad (Riquelme, 2006). Su utilidad viene dada porque se ha comprobado que los datos sin procesar tienen una utilidad mínima para los estudios. Esto se debe a que estos están llenos de vacíos, incongruencias, ruido, etc.

El Data Mining se encarga de extraer la información relevante para la toma de decisiones o la exploración de las mismas y ayuda a comprender el fenómeno que domina esa fuente de datos (Han, 2011).

De acuerdo a Pineda (2021), el procedimiento adecuado para el preprocesado de datos vendría a ser el siguiente:

4.2.1.1 Creación de conjuntos de entrenamiento y pruebas

Si la base de datos es extensa se recomienda separar en dos conjuntos: uno de entrenamiento y otro de pruebas. Esta división generalmente se hace de forma aleatoria pero siempre se consideran porcentajes aproximados a un 70% a 80% del total para el conjunto de entrenamiento.

4.2.1.2 Manejo de datos ausentes

Por lo general, después de recopilar los datos, es común notar que faltan algunos o muchos de ellos. Cuando se tiene este problema es necesario arreglar la base de datos puesto que la falta de datos puede llevar a resultados inesperados y confusos.

Para el manejo de estas situaciones se tienen tres opciones de los cuales enunciaremos dos:

- Eliminar las filas donde hay valores ausentes.
- Eliminar las columnas donde hay datos ausentes.

Estos procesos no siempre suelen ser factibles puesto que llevan a la pérdida de grandes conjuntos de información por lo que, especialmente en softwares como Python, se suele usar la técnica llamada imputación de valores ausentes que consiste en reemplazar el valor ausente con la media aritmética de toda la columna.

Otras opciones son sustituir el valor ausente por la mediana, el valor más frecuente o simplemente asignarle un valor constante.

4.2.1.3 Manejo de datos categóricos

Los algoritmos de aprendizaje trabajan con datos numéricos por lo que es necesario convertir los datos categóricos a datos de este tipo.

4.2.1.4 Escalamiento de características

El escalamiento es necesario porque los algoritmos de aprendizaje automático funcionan de una mejor forma cuando las características de un conjunto de datos poseen la misma escala.

Cabe recalcar que el autor recomienda profundizar más sobre el tema puesto que "el preprocesado de datos es la etapa de mayor importancia dentro del aprendizaje automático" (Riquelme, 2006).

4.2.1.5 Balanceo de datos

Puesto que los valores correspondientes a la alarma del sistema SCADA que representa la falla del módulo IGBT no son tan abundantes, se debe balancear para que estas se muestren con mucha más frecuencia. Esto se hace con el fin de que, al momento de ser usadas en el entrenamiento de los algoritmos, estos puedan contar con una mayor capacidad de detectarlas eficazmente.

Para este proceso hay muchos estudios tales como el de Hoyos (2019) o García (2021). Estos estudios prueban diversos tipos de balanceo dando un llamativo enfoque al denominado Subsampling en la clase mayoritaria; es decir, usar un algoritmo semejante a Random Forest para reducir la clase mayoritaria de datos y de esta forma obtener mejores resultados al entrenar el modelo.

4.2.2 Aprendizaje automático o Machine Learning

El aprendizaje automático o Machine Learning hace referencia a las herramientas informáticas o práctica de programación de computadoras que usan información pasada para tomar decisiones futuras (*López*, 2019).

El machine Learning se usa cuando se tenga problemas que necesitan una gran cantidad de variables para su solución puesto que las técnicas que se encuentran aquí simplifican trabajo u mejoran el desempeño. También se usa cuando no se pueda abordar el problema de forma tradicional o cuando el ambiente no es estable (Russell, 2018).

4.2.2.1 Tipos de machine Learning

Hay diferentes sistemas para Machine Learning que pueden ser clasificados de la siguiente manera (Russell, 2018):

4.2.2.1.1 Aprendizaje supervisado.

Este tipo de aprendizaje se caracteriza porque se entrena el algoritmo con datos que tienen la respuesta correcta, es por eso que mientras más grande sea la base de datos mejor serán los resultados.

Entre estos tipos de algoritmos se encuentran:

- K-nearest neigbors (KNN).
- Máquinas de vectores de soporte (SVN).
- Random Forest.
- Clasificadores Bayesianos.
- Regresión logística.
- Red neuronal.
- Arboles de decisiones.

4.2.2.1.2 Aprendizaje no supervisado.

Al contrario del aprendizaje supervisado, en el no supervisado no se tienen etiquetas por lo que el programa busca aprender por medio de la búsqueda de relaciones entre los datos de entrada basándose en alguna característica común (*Pineda*, 2021)

4.2.2.1.3 Aprendizaje por refuerzo.

Una inteligencia artificial ejecuta acciones por medio de la observación de un ambiente y de acuerdo a los resultados va aprendiendo por sí mismo.

4.2.3 Algoritmos a usar.

4.2.3.1 K-nearest neigbors (KNN)

Este algoritmo tiene un costo computacional alto y se desempeña mejor con bases de datos pequeñas.

El algoritmo se da por medio de los siguientes pasos:

- Calcular la distancia entre la muestra a clasificar y el resto de muestras del dataset.
- Seleccionar los K vecinos más cercanos con respecto a la muestra usando la Ecuación 1 distancia euclidiana.

$$Distancia_{\ euclidiana} = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \qquad (1)$$

También se puede usar la **Ecuación** 2 de distancia de Manhattan:

$$Distancia_{\ manhattan} = \sum_{i=1}^{n} |x_i - y_i| \qquad (2)$$

Existe otra ecuación llamada distancia de Minskowski (**Ecuación** 3):

$$Distancia_{\ minkowski} = \left(\sum_{i=1}^{n} |x_i - y_i|^{\ p} \right)^{\frac{1}{p}} \qquad (3)$$

En todas estas ecuaciones *x* e *y* son las muestras para el cálculo de la distancia mientras que *p* es una hiperparámetro que cambia su valor entre 1 y 2.

- Usando las ecuaciones anteriores se escoge la clase a la que pertenece la muestra de acuerdo a la clase que tienen sus vecinos más cercanos.

4.2.3.2 Máquinas de vectores de soporte (SVN)

Estas son un método muy efectivo usado para resolver problemas de clasificación. Consiste en la definición de un hiperplano o límite de decisión para dividir la muestra en dos grupos donde los datos que se ubiquen por encima de él serán consideradas positivas y las que se encuentren por debajo serán consideradas negativas. El objetivo fundamental en maximizar la distancia existente entre las muestras vecinas llamadas vectores de soporte e hiperplano separador. Esto se puede ver de mejor forma en la **Figura** 4:

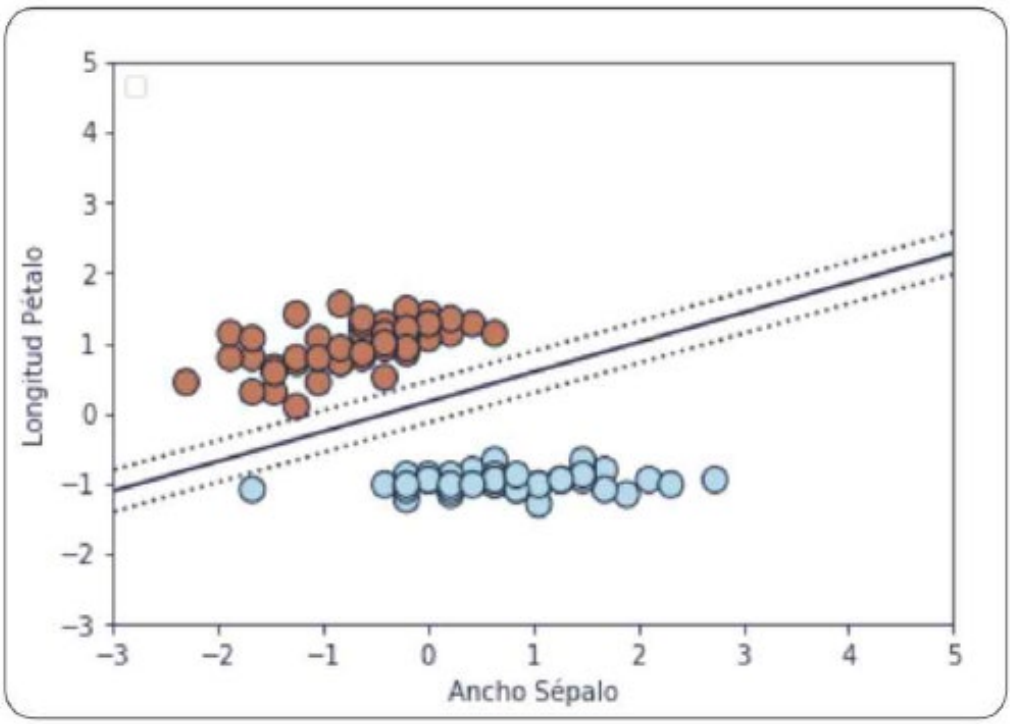


Figura 4**:** *Ejemplo del funcionamiento del algoritmo SVM.*

Fuente: Pineda, 2021.

De la Figura 4 se deduce que un punto x se ubicará encima del hiperplano si $wx_2 + b > 0$ se ubicará por debajo cuando sea menor que 0. Si $wx_2 + b = 0$ el punto se ubicará en el límite de decisión convirtiéndose en un vector de soporte

Se tiene que el margen es igual a la **Ecuación** 4:

$$Margen = X_2 - X_1 = \frac{2}{||w||} \quad (4)$$

El objetivo es reducir el margen 2/||w|| donde w es la norma euclidiana que puede ser calculada por medio de la **Ecuación** 5:

$$||W|| = \sqrt{W_o^2 + W_1^2 \dots W_n^2} \quad (5)$$

En lugar de maximizar el recíproco de la norma es mejor minimizar el cuadrado de la misma. El objetivo es lograr que sea derivable y permita la aplicación de algoritmos como el descenso de gradiente.

4.2.3.3 Random Forest

Los bosques aleatorios son un conjunto de árboles de decisión que tienen el fin de aumentar la exactitud en un modelo de clasificación. Se denominan así puesto que cada árbol es entrenado con un subconjunto de datos aleatorios y cada nodo toma un subconjunto aleatorio de atributos. Random forest toma su decisión basándose en la media de todas las predicciones.

4.2.3.3.1 Árboles de decisión.

Los árboles de decisiones nos permiten conocer las características más influyentes en un grupo de datos. Son famosos porque son representados en forma de árbol lo cual los hace fácil de entender.

Se componen de una serie de nodos que manejan reglas que deciden a partir de la evaluación de los valores de los atributos de las muestras. Dependiendo si estas están encima o debajo de un umbral son separadas en ramas o subconjuntos (López, 2019). El árbol se construye partiendo el conjunto de datos original en subconjuntos y volviéndose a partir de manera recursiva hasta que todas las muestras de una rama compartan la misma etiqueta.

La partición de un árbol en subconjuntos se efectúa con base en criterios de separación que dependen del tipo de dato: Discretos (**Figura** 5), continuos (**Figura** 6) o binarios (**Figura** 7).

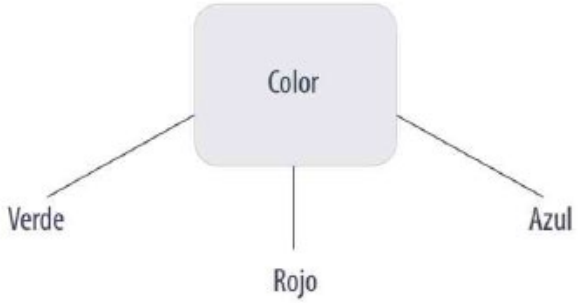


Figura 5**:** *Ejemplo de datos discretos*

Fuente: Carlos Pineda, 2021.

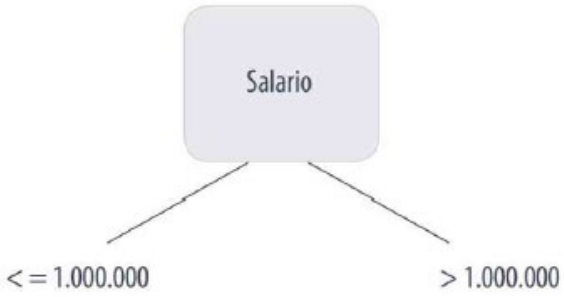


Figura 6**:** *Ejemplo de datos continuos.*

Fuente: Carlos Pineda, 2021

Figura 7: *Ejemplo de datos binarios.*

Fuente: Carlos Pineda, 2021.

4.2.3.4 Clasificador Bayesiano.

El clasificador Bayesiano también conocido como Clasificador ingenuo o Naive Bayes, es un algoritmo fácil de entrenar que puede ser usado en clasificaciones binarias como multiclase. Se lo considera ingenuo porque asume que las características en el conjunto de datos son independientes entre sí cuando visiblemente esto no es necesariamente cierto. Está basado en el teorema de Bayes.

Para hablar del teorema de Bayes se debe empezar con la probabilidad condicional. Esta consiste en determinar la probabilidad de que un evento A ocurra teniendo en cuenta que se ha dado un evento B. Esta probabilidad se calcula partiendo de dos sucesos A y B. Se representa de la siguiente manera (**Ecuación** 6):

$$P(A/B) = \frac{P(A \cap B)}{P(B)} \qquad (6)$$

Donde $P(A \cap B)$ es la probabilidad de que ocurran el evento A y el B.

El teorema de bayes es un aprendizaje basado en la experiencia porque es el cálculo de una probabilidad posterior a un evento A al tener en consideración eventos anteriores que pueden llevar a que dicho evento se dé.

La ecuación que representa el teorema de bayes es el siguiente (**Ecuación** 7):

$$(A/B) = \frac{P(B|A)P(A)}{P(B)} \tag{7}$$

Donde $P(B|A)$ es la probabilidad del evento B dada la ocurrencia de A, $P(A|B)$ es la probabilidad del evento A dado que ocurrió B y $P(B)$ es la probabilidad de que ocurran A y B (Pineda, 2021) (López, 2019).

4.3 Capítulo III: Métricas de rendimiento

Una vez entrenado el modelo de clasificación se debe evaluar que tan buenos o que tan malos son los resultados que arroja. Para ello se tienen técnicas especialmente dedicadas a métodos de clasificación

4.3.1 Matriz de confusión

Se trata de una cuadrícula que muestra la cantidad de aciertos y errores del algoritmo de clasificación. Este es exclusivo de los modelos de aprendizaje supervisado. Ver **Figura** 8.

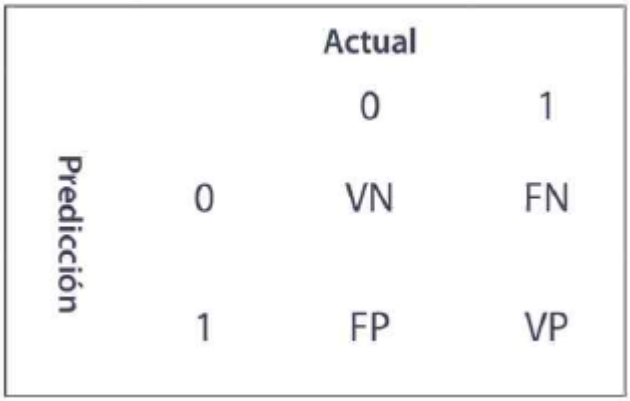


Figura 8**:** *Representación de una matriz de confusión*

Fuente**:** (Carlos Pineda, 2021.)

Donde:

- Verdaderos positivos (VP): cuando la clase real del punto de datos era 1 (Verdadero) y la predicha es también 1 (Verdadero).

- Verdaderos Negativos (VN): cuando la clase real del punto de datos fue 0 (Falso) y el pronosticado también es 0 (Falso).
- Falso Positivo (FP): cuando la clase real del punto de datos era 0 (False) y el pronosticado es 1 (True).
- Falso Negativos (FN): Cuando la clase real del punto de datos era 1 (Verdadero) y el valor predicho es 0 (Falso).

4.3.2 Exactitud

Este método mide el porcentaje de casos en los que se ha acertado por medio de la **Ecuación** 8:

$$Acurracy = \frac{VP + VN}{VP + VN + FP + FN} \tag{8}$$

4.3.3 Precisión

Es la relación entre las predicciones clasificadas correctamente como positivas y el número total de predicciones acertadas realizadas. Ver **Ecuación** 9.

$$Precisión = \frac{VP}{VP + VN} \tag{9}$$

4.3.4 Recall, sensibilidad o TPR.

Tasa de verdadero positivo, corresponde al número de elementos identificados correctamente como positivos verdaderos. Ver **Ecuación** 10

$$Recall = \frac{VP}{VP + FN} \tag{10}$$

4.3.5 F1

Esta métrica combina las medidas de precisión y sensibilidad en un solo valor siendo de gran utilidad cuando se quiera comprar el rendimiento combinado de ambas métricas. Ver **Ecuación** 11.

$$F1 = 2 * \frac{Precisión * Recall}{Precisión + Recall} \tag{11}$$

4.3.6 Tasa de falsos positivos

Esta métrica es la proporción entre los puntos de datos negativos que fueron considerados positivos erróneamente con respecto a los puntos negativos. Mientras más alto sea este valor significará que más muestras negativas fueron clasificadas positivas. Ver **Ecuación** 12.

$$FPR = \frac{FP}{FP + TN} \tag{12}$$

4.3.7 Curvas ROC

Estas curvas sirven para comparar diferentes clasificadores teniendo en cuenta sus predicciones que serán considerados entre 0 y 1. El eje x va a representar el incremento de la tasa de falsos positivos (especificidad), mientras el eje y representa la tasa de verdaderos positiv0os (sensibilidad). La línea representa un clasificador por lo que todas las curvas debajo de este umbral tienen un bajo desempeño mientras que las que estén por encima tienen un buen desempeño. Ver **Figura** 9.

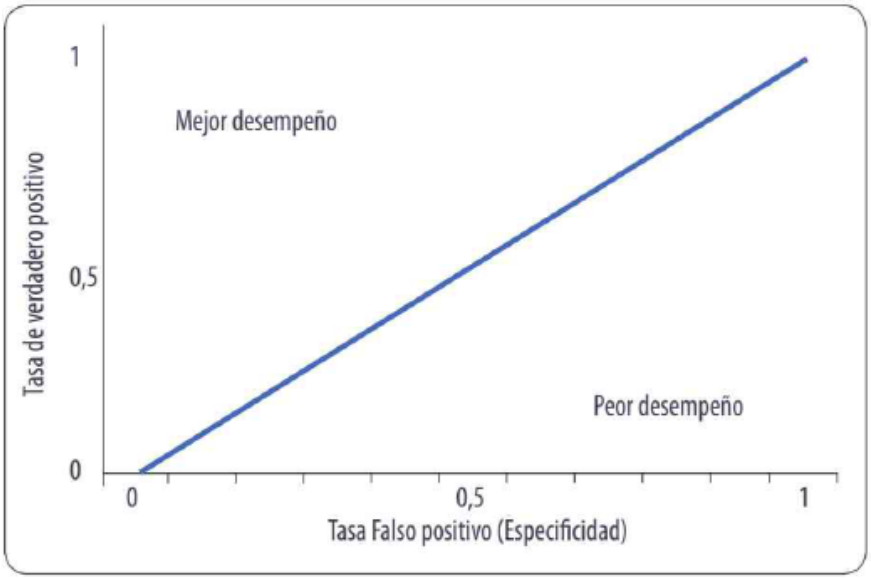


Figura 9**:** *Estructura de las curvas ROC.*

Fuente: **(**Pineda, 2021**)**

Las curvas ROC son especialmente útiles cuando se comparan varios clasificadores, ya que proporcionan una representación visual de cómo se desempeñan en términos de sensibilidad y especificidad para diferentes umbrales de decisión. La sensibilidad se refiere a la capacidad del modelo para identificar correctamente los casos positivos, mientras que la especificidad se refiere a la capacidad de evitar falsos positivos.

Al representar gráficamente la relación entre la sensibilidad y la especificidad, las curvas ROC permiten determinar cuál de los clasificadores es más efectivo en general. Un clasificador ideal tendría una curva ROC que se acercara lo más posible al rincón superior izquierdo del gráfico, lo que indicaría una alta sensibilidad y especificidad.

La línea diagonal en el gráfico representa un clasificador aleatorio que no tiene capacidad predictiva. Por lo tanto, cualquier curva ROC por encima de esta línea indica un

rendimiento mejor que el azar, mientras que las curvas por debajo indican un rendimiento peor que el azar.

5. Metodología

5.1 Área de estudio

5.1.1 Localización.

El parque eólico objeto de estudio se encuentra ubicado en la provincia de Loja y cantón Loja en las coordenadas XPXR+J83, a, Antigua Carretera Catamayo-Loja, Catamayo. Dicha ubicación se muestra en la **Figura** 10.

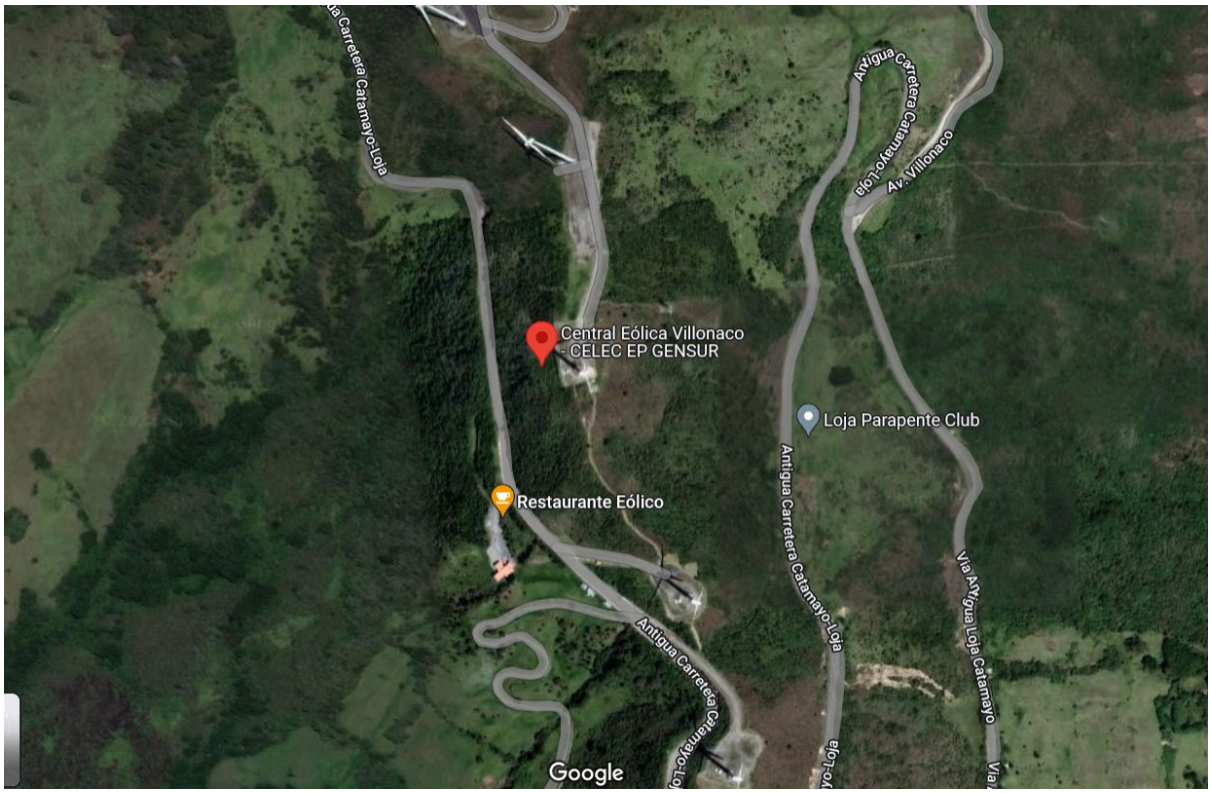


Figura 10: Ubicación del parque eólico Villonaco de acuerdo a Google Maps.

Fuente: (*Google Maps*, 2022)

5.1.2 Central eólica Villonaco

De acuerdo a la página de recursos y energía (Ministerio de Energía y Minas), esta central posee una potencia nominal de 16,5 MW proveniente de los 11 aerogeneradores marca GoldWind de 1,5 MW cada uno ubicados a lo largo de una distancia aproximada de 2 km. La central está instalada a una altura de 2700 msnm y cuenta con una velocidad promedio anual de 12,7 m/s. Su subestación de elevación 34,5 kV/69 kV tiene una capacidad de 25 MWA presentando un esquema de conexión de barra principal y transferencia.

5.1.3 Aerogenerador GoldWind GW70/1500

El aerogenerador de la marca GoldWind es de fabricación china. Fueron exportados vía marítima hacia la ciudad de Guayaquil, de donde fueron transportados a la ciudad de Loja vía terrestre.

Las características se muestran en la **Tabla** 2.

Tabla 2**:** *Características de los aerogeneradores GW70/1500.*

Parámetros de operación	**Especificaciones**
Potencia nominal	1500 KW
Velocidad de encendido	3 m/s
Velocidad de referencia del viento	11,8 m/s
Velocidad de parada	25 m/s (10 min)
Resistencia a velocidad de viento (3s)	70 m/s IEC-IA
Tiempo de vida útil	>= 20 años
Temperatura ambiente de operación	-30 °C a + 40 °C
Temperatura ambiente standby	-40 °C a + 50 °C
Rotor	**Especificaciones**
Diámetro	70 m
Área de barrido	3850 m^2
Rango de velocidad	10,2 a 19 rpm
Número de palas	3
Tipo de pala	LM34P o similar
Generador	**Especificaciones**
Tipo	Síncrono multipolar, generador con imán excitado permanente.
Potencia nominal	1500 Kw
Diseño	Accionamiento directo
Corriente nominal	660 A
Velocidad de rotación nominal	19 rpm
Clase de protección	F/IP23
Tipo de aislamiento	F
Convertidor	**Especificaciones**
Tipo	Convertidor IGBT
Clase de protección	IP54
rango del factor de potencia de salida	De -0,95 a +0,95
Voltaje nominal de salida	620/690 V
Corriente nominal de salida	1397/1255 A

Sistema de orientación	Especificaciones
Concepto de diseño	Mando por motor eléctrico
Movimiento nominal	0,5°/sec
Sistema de orientación	Freno 10 de retención
Sistema de freno	**Especificaciones**
Frenado aerodinámico	Triple hélice de paso
Freno mecánico	Sistema hidráulico
Torre	**Especificaciones**
Tipo	Metálica troncónica
Altura de buje	65 m
Diseño estándar	IEC 1024-I
Resistencia a tierra	<= 4Ω

Fuente:(Goldwind, 2022.)

5.2 Carta de confidencialidad.

La carta de confidencialidad puede ser contrada en el **Anexo** 1.

5.3 Enfoque metodológico – CRISP DM

Para este trabajo se hace uso de la metodología conocida como CRISP -DM. Esta metodología es una de las más populares a la hora de realizar proyectos de machine learning puesto que consta de varios pasos eficaces, fáciles de entender y progresivos que llevan a un resultado confiable. Estos pasos se pueden ver contrastados en los pasos mostrados en la **Figura** 12 **y Figura** 13**.**

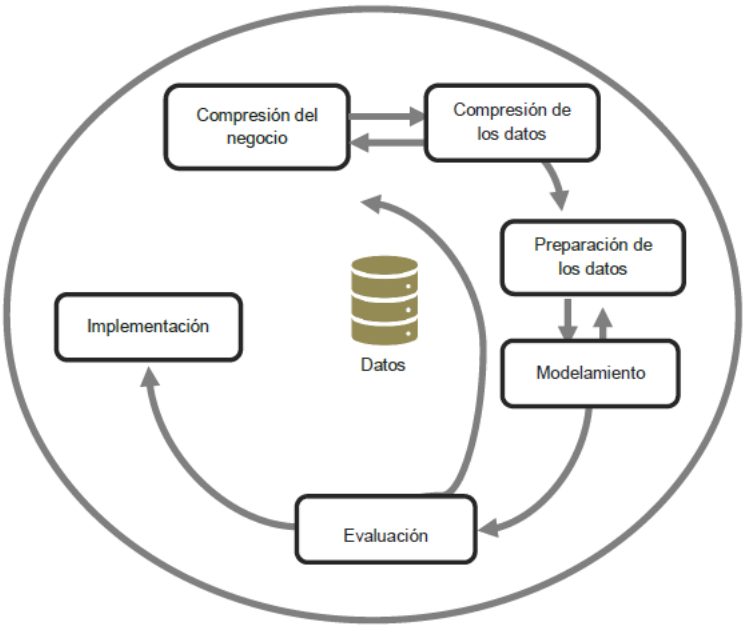


Figura 11: Estructura de la metodología CRISP DM

Fuente: GALÁN, 2016.

Figura 12: Flujo de tareas para el trabajo de Data Mining

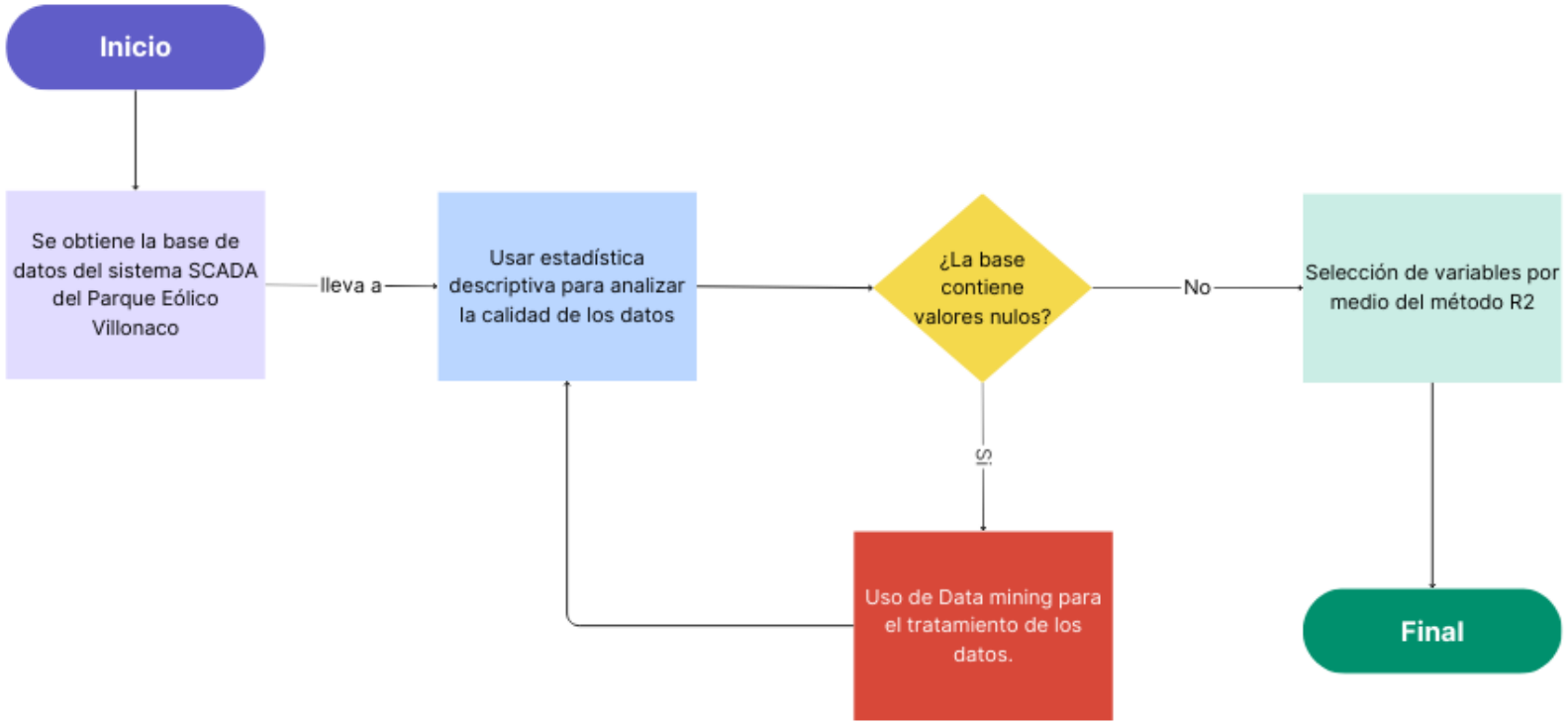
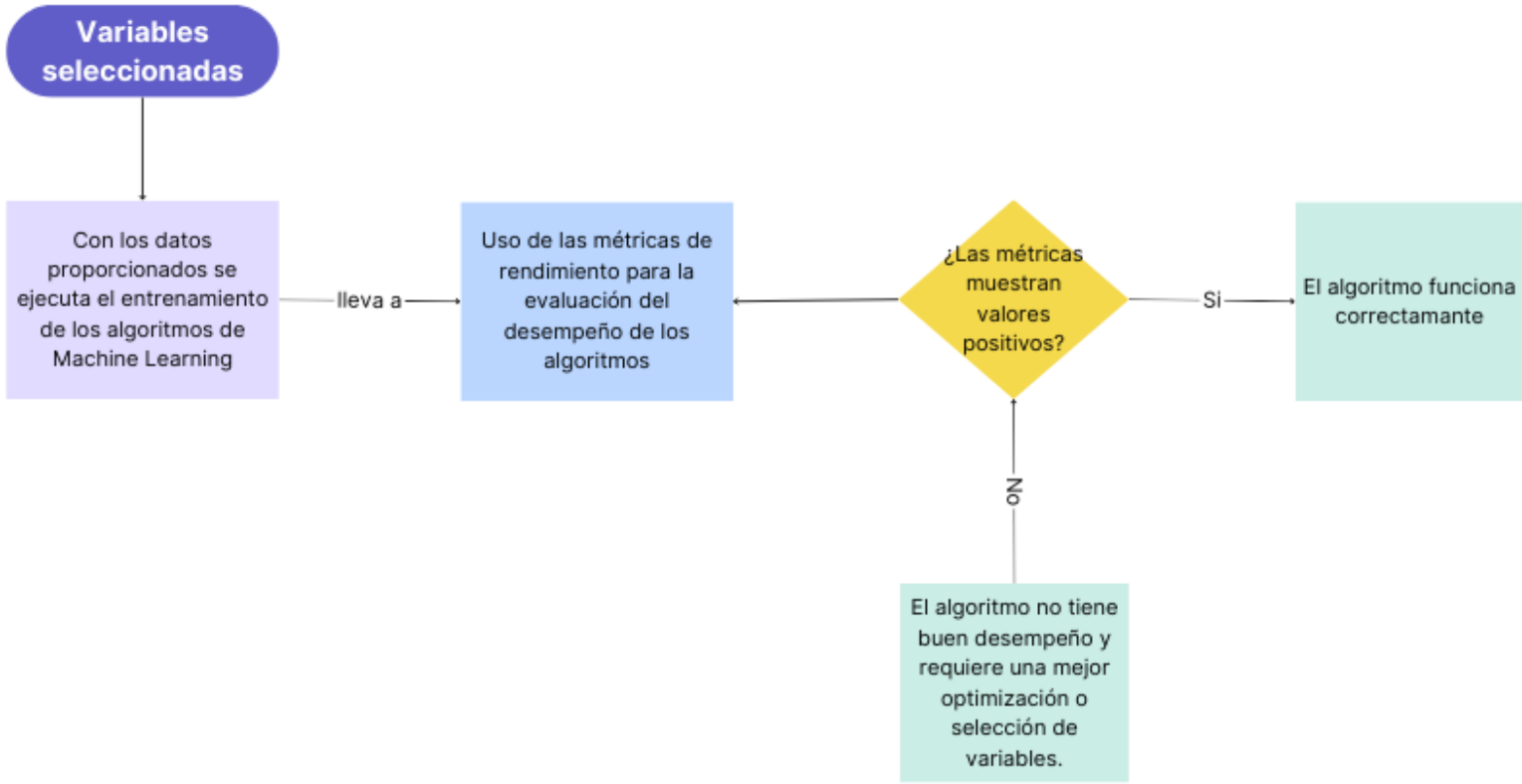


Figura 13: Flujo de tareas para entrenamiento de los algoritmos de machine Learning



5.3.1 Comprensión del negocio.

La aplicación de algoritmos de machine learning tiene como objetivo determinar si son factibles al momento de realizar predicciones de fallas en el módulo IGBT del convertidor de los aerogeneradores instalados en el parque eólico Villonaco, usando los datos provenientes del sistema SCADA de los mismos.

5.3.2 Contexto

La cantidad de fallas que se generan en el módulo IGBT son frecuentes y, debido a que se desconocen las probables causas de estas averías, es muy difícil determinar el momento o lapso de tiempo en que estos van cesar su funcionamiento.

5.3.3 Objetivos del negocio

Los objetivos del negocio es la correcta predicción de fallas en el módulo IGBT de acuerdo a los valores que toman las diferentes variables cuyos valores son recogidos por el sistema SCADA.

Los datos del sistema SCADA, al tener diferentes variables, pueden realizar predicciones de fallas de varios componentes, pero para este proyecto se han definido los siguientes objetivos:

- Usar técnicas de Data Mining para el preprocesamiento de los datos obtenidos por el sistema SCADA del parque eólico Villonaco.
- Aplicar los algoritmos de clasificación: KNN (Vecinos Cercanos), SVM (Máquinas de Soporte), RF (Random Forest) y NB (Clasificador Bayesiano) para la clasificación/detección de fallas.
- Evaluar y validar los algoritmos por medio de métricas de rendimiento para determinar el algoritmo con el mejor desempeño.

Estos objetivos pueden ayudar a la realización de mantenimiento predictivo a los aerogeneradores en casos para los cuales de detecte un fallo inminente. También puede ser usada como referencia en casos de estudios relacionados al fallo del módulo IGBT.

5.3.4 Criterios de éxito del negocio

Tratándose de un problema de machine learning, se establece como criterio de éxito que los algoritmos usados posean una alta fidelidad al momento de predecir fallas, el cual será determinado por medio de pruebas destinadas a valorar su desempeño al momento de realizar predicciones.

5.4 Evaluación de la situación

5.4.1 Base del sistema SCADA.

Se cuenta con un documento CSV con los datos recogidos por el sistema SCADA desde el 2014 hasta inicios del 2022. Esta base cuenta con un total de 70 variables y más de 4 millones de datos para cada una de ellas.

5.4.2 Recursos en cuanto a software

5.4.2.1 Python

De acuerdo al sitio de Amazon (2022), "Python es un lenguaje de programación ampliamente utilizado en las aplicaciones web, el desarrollo de software, la ciencia de datos y el machine learning. Los desarrolladores utilizan Python porque es eficiente y fácil de aprender, además de que se puede ejecutar en muchas plataformas diferentes".

El lenguaje Python fue desarrollado a finales de los años 80 por el programador holandés Guido Van Rossum para manejar excepciones y tener interfaces con Amoeba, que era el sistema operativo de la época. En el 2000 se lanzó Python 2.0 con nuevas características y en 2008 se lanzó Python 3.0.

5.4.2.2 SPSS

El programa Statical Product and Service Solutions, por sus siglas SPSS, es una herramienta que facilita el tratamiento estadístico de un conjunto de datos. Cuenta con una interfaz de ventanas y menús desplegables que hacen de este un programa muy fácil de usar.

Por medio de este programa se llevan a cabo tareas que van desde la creación de informes estadísticos de pequeños y grandes conjuntos de datos hasta la fabricación de imágenes estadísticas para una mejor apreciación de los resultados.

5.4.3 Recursos en cuanto a hardware.

Los recursos de hardware con los que se cuentan son una laptop cuyas características son las siguientes:

- Marca: Asus©.
- Modelo: X542UF – DM 342.
- Procesador: Intel(R) Core (TM) i5-8250U CPU @ 1.60GHz 1.80 GHz.
- Memoria ram: 8 GB.
- Capacidad de almacenamiento: 250 SSD – 1000 HDD.
- Tarjeta gráfica: NVIDIA GeForce MX130 v2GB.
- Sistema operativo: Windows 10 Pro.
- Monitor: LCD 15.6 FHD USLIM.

Sin embargo, el uso de Python se hará por medio Google Colab, el cual es un producto de Google Research que permite ejecutar código de Python en el navegador de nuestra preferencia.

Los recursos que ofrece Google Colab varían con el tiempo en su versión gratuita, pero al momento de la ejecución de este trabajo cuenta con CPU Intel Xeon a 2.20 GHz, 13 GB de RAM, acelerador Tesla K80 y 12 GB de VRAM GDDR5.

5.4.4 Base de datos.

La base de datos a usar será la proporcionada por la empresa CELEC EP. gracias al convenio educativo con la Universidad Nacional de Loja. Esta cuenta con los datos recogidos por el sistema SCADA de la central eólica Villonaco.

5.4.5 Costes y beneficios.

Este proyecto no requiere costos adicionales puesto que la empresa CELEC cuenta con un convenio con la Universidad Nacional de Loja por lo que los datos fueron proporcionados para fines educativos.

En cuanto a beneficios económicos no se supone ninguno para la Universidad Nacional de Loja, pero en caso de implementarse en la central eólica para un monitoreo en tiempo real, podría evitar la falla de los módulos IGBT y así reducir los costos causados por tener que reemplazar estos elementos y el tiempo que los aerogeneradores deben permanecer fuera de funcionamiento mientras se realiza la reparación.

5.5 Comprensión de los datos.

Esta es la segunda parte de la metodología CRISP – DM. Esta parte es la encargada de estudiar la calidad de los datos.

La base de datos con la que se cuenta, ordena sus datos respecto a una fecha. Dada la naturaleza del parque, existen lapsos de tiempo en los que no se recogió información de una o varias variables por lo que, en la base de datos, estos momentos se ven representados con un valor nulo. De acuerdo a Pineda (2021), si este hecho no es tratado adecuadamente pueden provocar errores o resultados incongruentes en los modelos de aprendizaje automático.

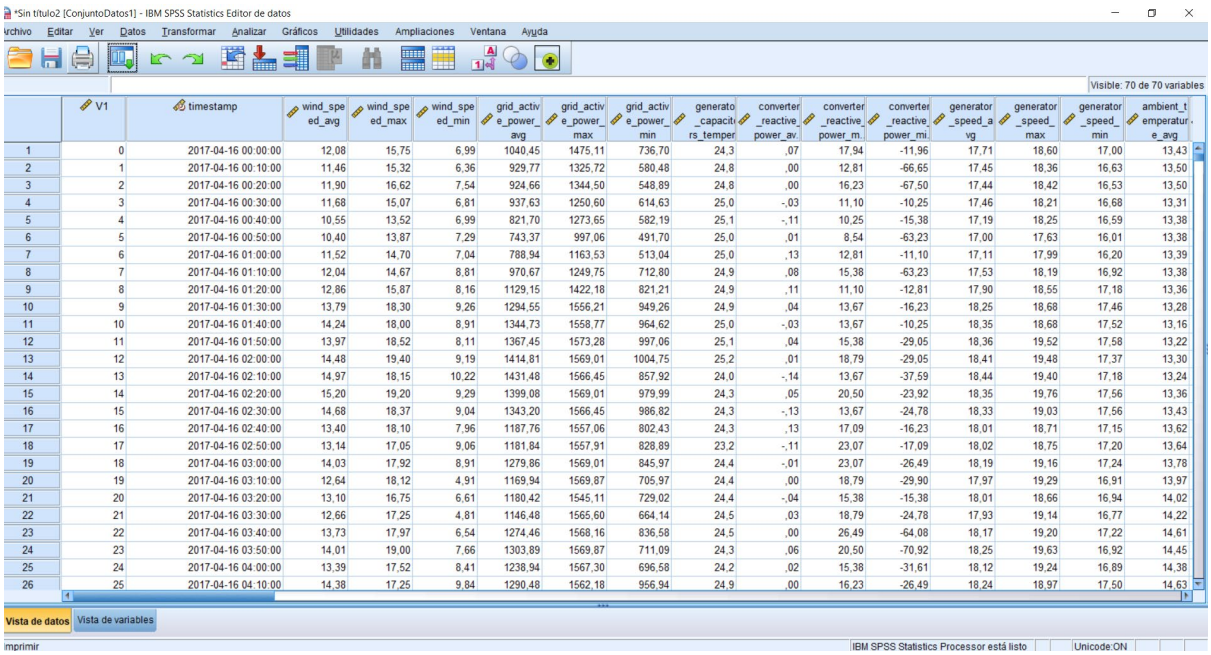
Para realizar este trabajo se carga la base de datos en Google Drive puesto que Google Colab tiene la capacidad de leer documentos directamente desde este sitio. En caso de no realizar este paso se deberá cargar el documento desde el almacenamiento interno de la computadora y, dado su gran tamaño, esto puede tomar mucho tiempo y retrasar considerablemente el trabajo puesto que este procedimiento debe realizarse cada vez que se abra el entorno de Python.

5.5.1 Técnicas

Para averiguar el estado inicial de la base de datos se trabajó simultáneamente en Python y SPSS para realizar una observación general de la base y comprobar la calidad de los datos.

5.5.2 Datos y tamaño de la base de datos.

Al usar SPSS y cargar el archivo CSV se mostrará una ventana en la que se pueden visualizar los datos (**Figura** 14); sin embargo, es en la pestaña desplegable Analizar – Estadísticos descriptivos – Frecuencias, en donde se procederá inicialmente.



	V1	timestamp	wind_speed_avg	wind_speed_max	wind_speed_min	grid_active_power_avg	grid_active_power_max	grid_active_power_min	generator_capacity	converter_reactive_power_av	converter_reactive_power_m	converter_reactive_power_mi	generator_speed_avg	generator_speed_max	generator_speed_min	ambient_temperature_avg
1	0	2017-04-16 00:00:00	12,08	15,75	6,99	1040,45	1475,11	736,70	24,3	,07	17,94	-11,96	17,71	18,60	17,00	13,43
2	1	2017-04-16 00:10:00	11,46	15,32	6,36	929,77	1325,72	580,48	24,8	,00	12,81	-66,65	17,45	18,36	16,63	13,50
3	2	2017-04-16 00:20:00	11,90	16,62	7,54	924,66	1344,50	548,89	24,8	,00	16,23	-67,50	17,44	18,42	16,53	13,50
4	3	2017-04-16 00:30:00	11,68	15,07	6,81	937,63	1250,60	614,63	25,0	-,03	11,10	-10,25	17,46	18,21	16,68	13,31
5	4	2017-04-16 00:40:00	10,55	13,52	6,99	821,70	1273,65	582,19	25,1	-,11	10,25	-15,38	17,19	18,25	16,59	13,38
6	5	2017-04-16 00:50:00	10,40	13,87	7,29	743,37	997,06	491,70	25,0	,01	8,54	-63,23	17,00	17,63	16,01	13,38
7	6	2017-04-16 01:00:00	11,52	14,70	7,04	788,94	1163,53	513,04	25,0	,13	12,81	-11,10	17,11	17,99	16,20	13,39
8	7	2017-04-16 01:10:00	12,04	14,67	8,81	970,67	1249,75	712,80	24,9	,08	15,38	-63,23	17,53	18,19	16,92	13,38
9	8	2017-04-16 01:20:00	12,86	15,87	8,16	1129,15	1422,18	821,21	24,9	,11	11,10	-12,81	17,90	18,55	17,18	13,36
10	9	2017-04-16 01:30:00	13,79	18,30	9,26	1294,55	1556,21	949,26	24,9	,04	13,67	-16,23	18,25	18,68	17,46	13,28
11	10	2017-04-16 01:40:00	14,24	18,00	8,91	1344,73	1558,77	964,62	25,0	-,03	13,67	-10,25	18,35	18,68	17,52	13,16
12	11	2017-04-16 01:50:00	13,97	18,52	8,11	1367,45	1573,28	997,06	25,1	,04	15,38	-29,05	18,36	19,52	17,58	13,22
13	12	2017-04-16 02:00:00	14,48	19,40	9,19	1414,81	1569,01	1004,75	25,2	,01	18,79	-29,05	18,41	19,48	17,37	13,30
14	13	2017-04-16 02:10:00	14,97	18,15	10,22	1431,48	1566,45	857,92	24,0	-,14	13,67	-37,59	18,44	19,40	17,18	13,24
15	14	2017-04-16 02:20:00	15,20	19,20	9,29	1399,08	1569,01	979,99	24,3	,05	20,50	-23,92	18,35	19,76	17,56	13,36
16	15	2017-04-16 02:30:00	14,68	18,37	9,04	1343,20	1566,45	986,82	24,3	-,13	13,67	-24,78	18,33	19,03	17,56	13,43
17	16	2017-04-16 02:40:00	13,40	18,10	7,96	1187,76	1557,06	802,43	24,3	,13	17,09	-16,23	18,01	18,71	17,15	13,62
18	17	2017-04-16 02:50:00	13,14	17,05	9,06	1181,84	1557,91	828,89	23,2	-,11	23,07	-17,09	18,02	18,75	17,20	13,64
19	18	2017-04-16 03:00:00	14,03	17,92	8,91	1279,86	1569,01	845,97	24,4	-,01	23,07	-26,49	18,19	19,16	17,24	13,78
20	19	2017-04-16 03:10:00	12,64	18,12	4,91	1169,94	1569,87	705,97	24,4	,00	18,79	-29,90	17,97	19,29	16,91	13,97
21	20	2017-04-16 03:20:00	13,10	16,75	6,61	1180,42	1545,11	729,02	24,4	-,04	15,38	-15,38	18,01	18,66	16,94	14,02
22	21	2017-04-16 03:30:00	12,66	17,25	4,81	1146,48	1565,60	664,14	24,5	,03	18,79	-24,78	17,93	19,14	16,77	14,22
23	22	2017-04-16 03:40:00	13,73	17,97	6,54	1274,46	1568,16	836,58	24,5	,00	26,49	-64,08	18,17	19,20	17,22	14,61
24	23	2017-04-16 03:50:00	14,01	19,00	7,66	1303,89	1569,87	711,09	24,3	,06	20,50	-70,92	18,25	19,63	16,92	14,45
25	24	2017-04-16 04:00:00	13,39	17,52	8,41	1238,94	1567,30	696,58	24,2	,02	15,38	-31,61	18,12	19,24	16,89	14,38
26	25	2017-04-16 04:10:00	14,38	17,25	9,84	1290,48	1562,18	956,94	24,9	,00	16,23	-26,49	18,24	18,97	17,50	14,63

Figura 14*: Ventana de visualización de SPSS*

Al seguir dicha dirección se despliega una ventana en la cual se deberán escoger las variables de las que se desea hacer el análisis (**Figura** *15*). En este caso se seleccionan todas las variables a excepción de las variables “V1”, “timestamp” y “aeronumber” puesto que la primera es la numeración de las filas de cada variable, la segunda corresponde a la serie de tiempo y la tercera hace referencia al número de aerogenerador.

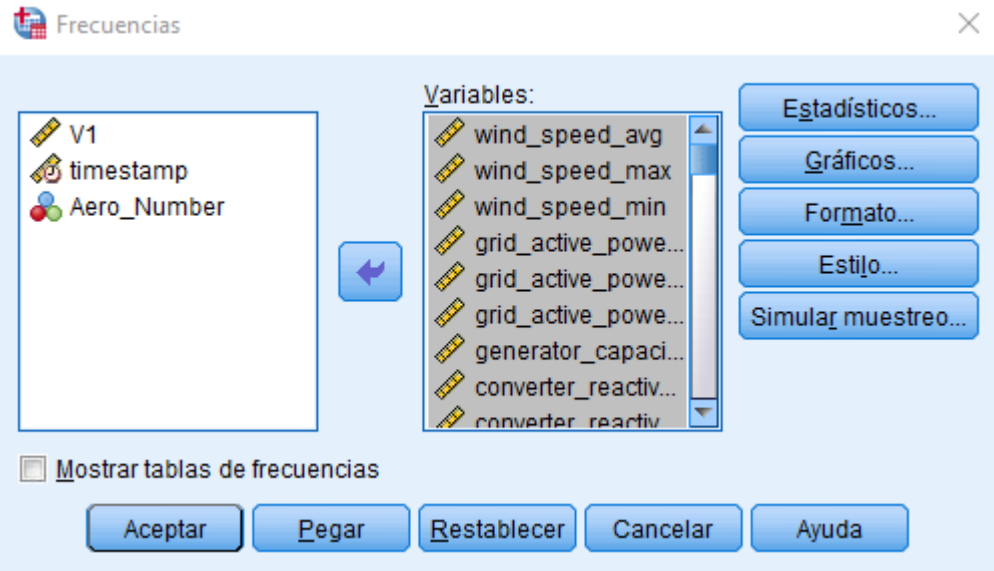


Figura 15**:** *Ventana de trabajo dentro del software SPSS con las variables seleccionadas.*

Una vez se hayan seleccionado las variables se procede a entrar en la opción Estadísticos (**Figura** *16*). En esta ventana marcan aquellas opciones que se crean necesarias para la evaluación de la calidad de datos. Se da click en continuar.

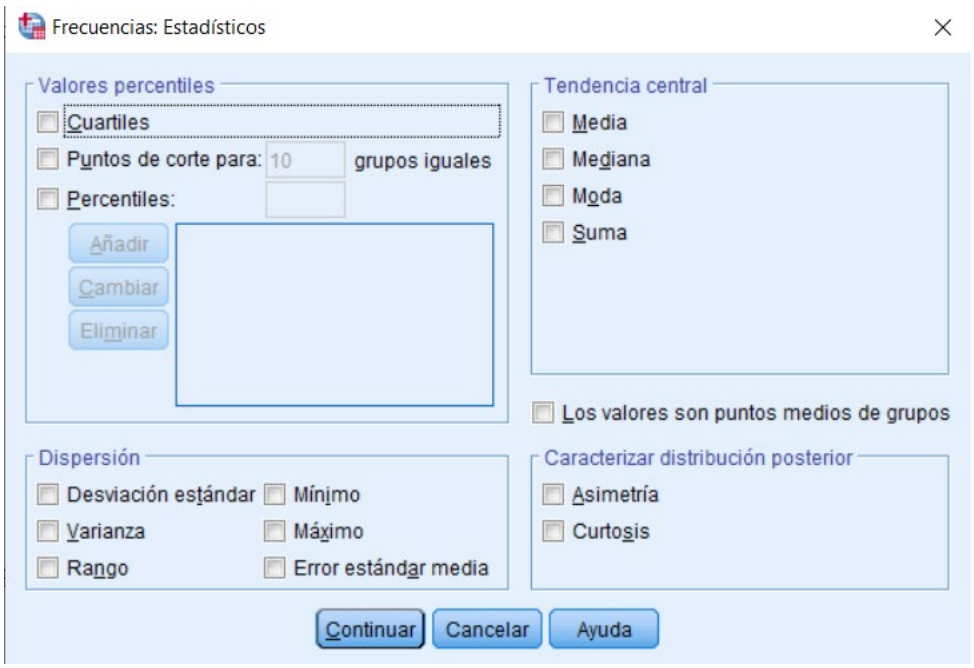


Figura 16**:** *Opción para crear informes estadísticos en el software SPSS.*

Seguidamente, en la opción gráficos, se selecciona la opción histograma (**Figura** *17*). Con esta combinación se obtienen datos relevantes para averiguar la condición en que se encuentran los datos de las variables de la base original, además del conteo de sus valores válidos y perdidos, es decir, datos nulos.

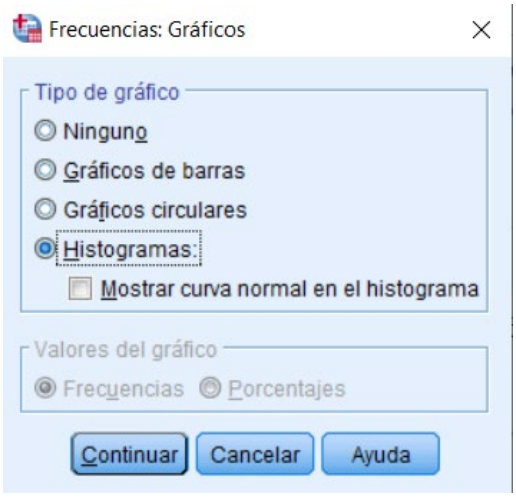


Figura 17**:** *Opción para crear histogramas en SPSS.*

Debido a las capacidades limitadas de la máquina, este proceso puede llevar varios minutos por lo que también se realizó un proceso similar en Python.

Por medio de este entorno se obtuvo el porcentaje de datos perdidos en cada una de las columnas de la base; es decir, el porcentaje de datos perdidos de cada variable. Con este dato se puede decidir cómo proceder para el manejo de datos faltantes.

5.5.3 Manejo de datos ausentes:

Para el manejo de los datos ausentes se usará la metodología mencionada en Pineda (2021).

5.5.3.1 Columnas con datos faltantes.

De acuerdo a Pineda (2021), en caso de comprobarse que existan datos nulos, existen varias maneras de tratarlos: La primera de ellas es reemplazar este valor por la media aritmética de toda la fila y la segunda es eliminar toda la fila donde se encuentren estos datos faltantes. Sin embargo, también cita que esta última opción no es muy recomendable debido a que se podrían perder muchos datos que podrían ser útiles para el aprendizaje automático.

En caso de que la cantidad de datos faltantes en una columna sea insignificante en comparación con el tamaño que se tiene, la opción más simple y factible es la eliminación de las filas en que se encuentran, por otra parte, si es demasiado alto se procederá a eliminar la columna entera puesto que, si se decide rellenar los datos faltantes con el valor medio de los demás datos, se va a crear una tendencia en la que ese valor va a repetirse muchas veces y va a afectar al entrenamiento de los modelos de machine learning (Pineda, 2021).

5.5.3.2 Manejo de filas con datos nulos.

En caso de las filas también se recomienda reemplazar los valores nulos por la media en caso de que el porcentaje sea insignificante, aunque el eliminar dichos datos también es una

opción válida. Dado que se está trabajando con una serie de tiempo, si se detecta que existe una ausencia de datos durante un periodo de tiempo significativo, se recomienda trabajar con el lapso de tiempo más largo en el que se cuente con datos (Pineda, 2021).

5.5.4 Ordenar datos por fecha.

Dado que se trabaja con una serie de tiempo, se debe verificar que los datos estén ordenados correctamente para que la secuencia de los datos se a la correcta al momento de entrenar el algoritmo de machine learning. Además, se debe realizar esta verificación para que al momento de obtener los diagramas de dispersión se pueda observar si existen lapsos de tiempos en los que haya ausencia de datos.

5.5.5 Selección de variables.

Para la selección de variables se usa el método del coeficiente de determinación R2, el cual mide la relación que guardan las variables independientes con la variable dependiente la cual, en este caso, es la falla del módulo IGBT.

Este método se realiza por medio del software Python y se toman las variables cuya relación sea superior al 5%. Las demás serán descartadas puesto que su uso durante la predicción de los datos será poco relevante y hasta perjudicial puesto que aumentarían el tiempo de entrenamiento de los algoritmos, además de causar sobre entrenamiento y resultados dudosos.

Para ello se usa la librería de Python funpymodeling que sirve para hacer análisis univariados numéricos dentro de este software. El código es relativamente simple y la creación del nuevo documento de Excel en el que consten las variables seleccionadas se ejecuta dentro de este mismo.

5.6 Aplicación de algoritmos de machine learning.

Para la aplicación de algoritmos de machine learning no es necesario poseer un conocimiento profundo del software Python puesto que existe mucha documentación en la que ese detalla el código con el que se puede realizar este trabajo y, en caso de realizarlo en Google Colab, todas las librerías necesarias están disponibles, por lo que se evita el tener que instalar ambientes y librerías extra.

La librería “SCIKIT-LEARN” será usada repetidamente ya que ella se encuentran las herramientas necesarias para programar, correr, graficar y validar los algoritmos de machine learning propuestos en este trabajo.

A continuación, se procede a explicar el procedimiento usado para cada uno de los algoritmos planteados.

5.6.1 Random Forest

Dentro del entorno de Python se importan las librerías necesarias y los datos para el entrenamiento. En este caso se usaron las librerías “NUMPY”, “MATPLOTLIB”, “PANDAS” y “SKLEARN”; mientras que para los datos se usaron los obtenidos en la selección de variables. Estos datos son guardados en un dataframe (**Figura** *18*).

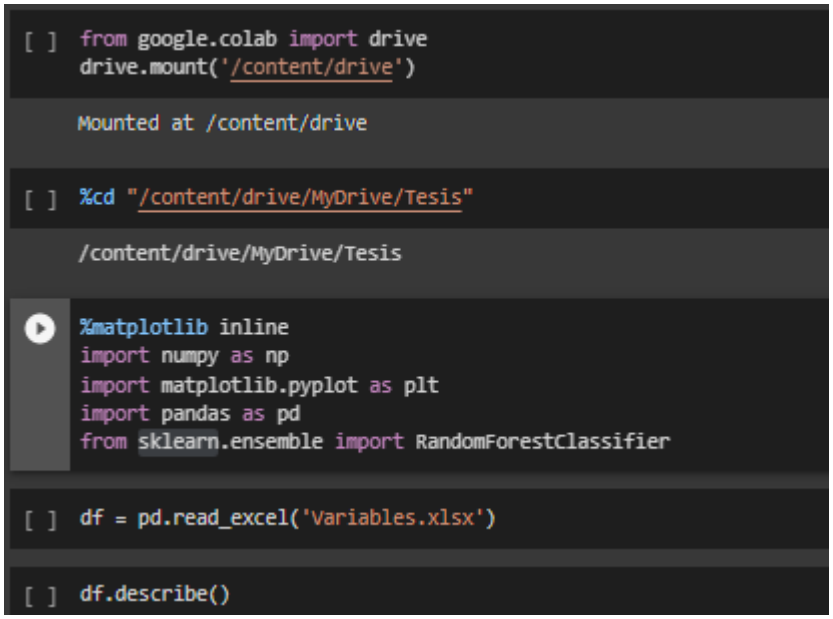


Figura 18**:** *Carga de datos y librerías al entorno de Python para el algoritmo de Random Forest.*

Inicialmente se verifica el estado de los datos con la función *describe* que muestra información estadística de cada una de las columnas que conforman el dataframe y la cantidad de filas y columnas que lo conforman. Con esta información se revisa que las columnas correspondan a las variables seleccionadas.

Con la función *info* se revisa nuevamente que no existan datos nulos y que todas las variables sean del mismo tipo de datos. Para este caso deberán ser tipo float64.

```
[ ] df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 5551 entries, 0 to 5550
Data columns (total 24 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   Unnamed: 0                            5551 non-null   int64
 1   wind_speed_avg                        5551 non-null   float64
 2   wind_speed_max                        5551 non-null   float64
 3   wind_speed_min                        5551 non-null   float64
 4   grid_active_power_avg                 5551 non-null   float64
 5   grid_active_power_max                 5551 non-null   float64
 6   grid_active_power_min                 5551 non-null   float64
 7   grid_I1_avg                           5551 non-null   float64
 8   grid_I2_avg                           5551 non-null   float64
 9   grid_I3_avg                           5551 non-null   float64
 10  winding_temperature_max               5551 non-null   float64
 11  igbt_temperature_max                  5551 non-null   float64
 12  dc_link_capacitors_temperature_max    5551 non-null   float64
 13  ac_inductor_temperature_max           5551 non-null   float64
 14  rectifier_temperature_max             5551 non-null   float64
 15  dc_inductor_temperature_max           5551 non-null   float64
 16  step_up_igbt_temperature_max          5551 non-null   float64
 17  pitch_motor_temperature_1_max         5551 non-null   float64
 18  pitch_motor_temperature_2_max         5551 non-null   float64
 19  pitch_motor_temperature_3_max         5551 non-null   float64
 20  pitch_converter_temperature_1_max     5551 non-null   float64
 21  pitch_converter_temperature_2_max     5551 non-null   float64
 22  pitch_converter_temperature_3_max     5551 non-null   float64
 23  X                                     5551 non-null   int64
dtypes: float64(22), int64(2)
memory usage: 1.0 MB
```

Figura 19**:** *Revisión preventiva de los datos del dataframe.*

Dado que en la **Figura** 19 se observa que existe una columna llamada Unnamed:0 de clase int64, se procede a verificar sus datos. Por medio de esto se nota que esta columna no es más que un conteo de los datos totales por lo que se elimina dicha variable con todos sus valores.

Una vez se comprueben los datos, se dividen en datos para entrenamiento y prueba con la herramienta train_test de la librería SKLEARN. La división se da por defecto en un 75% de los datos para entrenamiento y 25% para prueba. Este valor puede ser cambiado, pero para este trabajo se trabaja con los valores por defecto.

Se crea una variable en la que se definen los hiperparámetros del algoritmo. El más importante de ellos es el número de árboles. Este valor es decidido luego de varias pruebas. Adicionalmente se define el parámetro min_samples_leaf que representa el número mínimo de muestras que debe haber en un nodo final. Con estos ajustes se tiene listo el algoritmo y se envía a entrenar.

5.6.2 SVM

Para este algoritmo se usan las mismas librerías y los mismos datos que los usados para entrenar el algoritmo de Random Forest (Figura *16*), la diferencia radica en la herramienta de sklearn que será usada. Para este algoritmo se usará la librería sklearn.svm y la herramienta SVC.

Los pasos a seguir son idénticos a los dados en el algoritmo anterior exceptuando la asignación de los hiperparámetros. Para este algoritmo se definirán de la siguiente manera (**Figura** *20*):

```
modelo = SVC(C = 100, kernel = 'linear', random_state=123)
modelo.fit(X_train, y_train)
```

Figura 20**:** *Asignación de hiperparámetros para algoritmo SVM.*

Cabe recalcar que el tiempo de entrenamiento de este algoritmo es muy superior al de Random Forest por lo que, al obtener resultados se debe determinar si vale la pena entrenar este algoritmo.

Existen dos tipos principales de SVM (lineal y radial), pero dada la naturaleza de la investigación solo se entrenará el algoritmo para SVM lineal.

5.6.3 KNN

El procedimiento de carga de librerías y datos es exactamente el mismo que en los casos anteriores a excepción de que la librería a usar será sklearn.neighbors y la herramienta KNeighborsClassifier (**Figura** *21*).

```
[ ]  import pandas as pd
     import numpy as np
     import matplotlib.pyplot as plt
     from matplotlib.colors import ListedColormap
     import matplotlib.patches as mpatches
     import seaborn as sb
     from sklearn.model_selection import train_test_split
     from sklearn.preprocessing import MinMaxScaler
     from sklearn.neighbors import KNeighborsClassifier
     from sklearn.metrics import classification_report
     from sklearn.metrics import confusion_matrix
```

Figura 21**:** *Carga de datos y librerías al entorno de Python para el algoritmo de vecinos cercanos.*

Uno de los hiperparámetros más importantes de este algoritmo es el número de vecinos que se usarán para el entrenamiento. Este dato puede ser obtenido por medio de la ejecución del código de la **Figura** 22.

```
k_range = range(1, 20)
scores = []
for k in k_range:
 knn = KNeighborsClassifier(n_neighbors = k)
 knn.fit(X_train, y_train)
 scores.append(knn.score(X_test, y_test))
plt.figure()
plt.xlabel('k')
plt.ylabel('accuracy')
plt.scatter(k_range, scores)
plt.xticks([0,5,10,15,20])
```

Figura 22**:** *Código para la ejecución del algoritmo KNN.*

Este código hace un barrido desde 1 a 20 vecinos cercanos, evalúa la precisión obtenida y genera un gráfico en el que muestra los resultados.

Una vez obtenido el número de vecinos cercanos óptimos se ajusta ese hiperparámetro en las líneas de código para el entrenamiento (**Figura** 23).

```
n_neighbors = 8
knn = KNeighborsClassifier(n_neighbors)
knn.fit(X_train, y_train)
print('Accuracy of K-NN classifier on training set: {:.2f}'
     .format(knn.score(X_train, y_train)))
print('Accuracy of K-NN classifier on test set: {:.2f}'
     .format(knn.score(X_test, y_test)))
```

Figura 23**:** *Código para el entrenamiento del algoritmo KNN en el entorno de Python.*

En la Figura *21* se pueden observar varios *print*, estos sirven para mostrar la eficacia que se tuvo al entrenar el algoritmo.

5.6.4 Naive Bayes

Al igual que en los anteriores algoritmos, las librerías usadas son las mismas a excepción de sklearn.naive_bayes (**Figura** 24). De igual forma la carga de datos se realiza de la misma forma que en los casos anteriores.

```
[ ]  from sklearn.naive_bayes import GaussianNB
     classifier = GaussianNB()
     classifier.fit(X_train, y_train)
```

Figura 24: *Exportación de librerías para implementación del algoritmo de Naive Bayes.*

5.6.5 Evaluación de los resultados

Para la evaluación de los resultados se hizo uso de las métricas mencionadas en el marco teórico del presente trabajo. Para estas métricas es necesario la importación de diferentes herramientas (**Figura** *25*).

```
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
```

Figura 25: *Importación de herramientas para la obtención de las métricas para machine learning.*

La herramienta classificatión report permite obtener las siguientes métricas de precisión: recall, f1 score y support. Estos valores se obtienen por medio de la aplicación del código que consta en la **Figura** *26*.

```
pred = BA_model.predict(X_test)
print(confusion_matrix(y_test, pred))
print(classification_report(y_test, pred))
```

Figura 26: *Código requerido para la obtención de las métricas de rendimiento.*

Para la obtención de la métrica Acurracy se usa el comando score que devuelve el valor numérico requerido.

Una vez se tenga el modelo entrenado se aplicó el código que se muestra en la **Figura** *27*, donde BA_model es el modelo entrenado.

```
# Accuracy promedio
BA_model.score(X_test, y_test)
```

Figura 27: *Código requerido para obtener el Acurracy.*

5.7 Curvas ROC

Para la obtención de las curvas ROC (AUC por sus siglas en inglés), se debe obtener el valor bajo la curva. Esto se hace por medio del software Python y la importación de nuevas herramientas que se pueden visualizar en la **Figura** *28*.

```
from sklearn.metrics import roc_curve
from sklearn.metrics import roc_auc_score
```

Figura 28: *Importación de herramientas para la curva ROC*

Una vez importadas las herramientas se debe definir la curva ROC por medio del código mostrado en la Figura *29*.

```
def plot_roc_curve(fpr, tpr):
    plt.plot(fpr, tpr, color='orange', label='ROC')
    plt.plot([0, 1], [0, 1], color='darkblue', linestyle='--')
    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')
    plt.title('Receiver Operating Characteristic (ROC) Curve')
    plt.legend()
    plt.show()
```

Figura 29: Forma de definir la curva ROC.

A continuación, se definen las variables que se usarán en la curva y se obtiene el valor AUC (**Figura** *30***)**

```
[ ]  probs = modelo.predict_proba(X_test)

[ ]  probs = probs[:, 1]

[ ]  auc = roc_auc_score(y_test, probs)
     print('AUC: %.2f' % auc)
```

Figura 30: Definición de variables para la curva ROC

Finalmente se ejecutan el comando mostrado en la
Figura 31.

```
fpr, tpr, thresholds = roc_curve(y_test, probs)

plot_roc_curve(fpr, tpr)
```

Figura 31: Comando para generar la curva ROC

6. Resultados

6.1 Análisis y preparación de los datos

6.1.1 Estadística descriptiva de la base original

Una vez procesados los datos, el software SPSS arrojó los datos que se pueden encontrar en la **Tabla** 3. Dada la cantidad de datos y columnas solamente se muestran algunas de las columnas tomadas al azar mientras que en la **Figura** *26* se muestra un gráfico que representa la cantidad total de datos válidos y nulos.

Tabla 3: *Muestra de la estadística descriptiva de la base original.*

Variable	wind_speed_avg	wind_speed_max	data_power_production_time	two_day_before_data_energy_yield	three_days_before_data_energy_yield
Número de datos nulos	0	0	2	1820.948	1820.948
Media	10,2970	13,4731	34.535,8117	19.095,062	19.056,043
Mediana	10,4900	13,6900	33.779,81	19,607	19,552
Moda	0,00	0,00	9.833,07	0,0	0,0
Desviación	5,00	6,11	25.204,84	12.306,84	12.310,93
Varianza	25,002	37,340	635284.291,9	151458.549,5	151559.210,4
Rango	40,97	40,97	795.691,57	61,844	61,844
Mínimo	0,00	0,00	12,83	0,0	0,0
Máximo	40,97	40,97	795.704,40	61,844	61,844

.

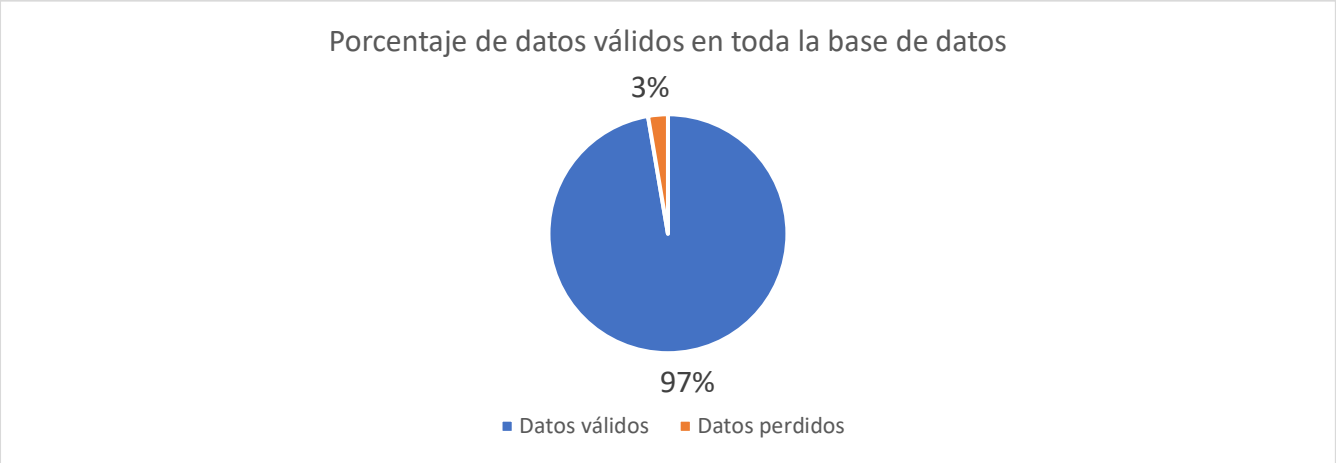


Figura 32**:** *Porcentaje de datos válidos en toda la base de datos.*

Tal y como se muestra en la **Figura** 32, existen columnas cuyo valor de datos perdidos es nulo, otras en las que es muy leve y otras columnas en las que se puede encontrar un gran porcentaje de datos nulos. Se muestra que existen variables que contienen datos nulos en diferentes proporciones. De estas se localizaron cuatro variables cuyo porcentaje de datos ausentes es muy alto. Esta falta de datos se muestra en la **Figura** *33*.

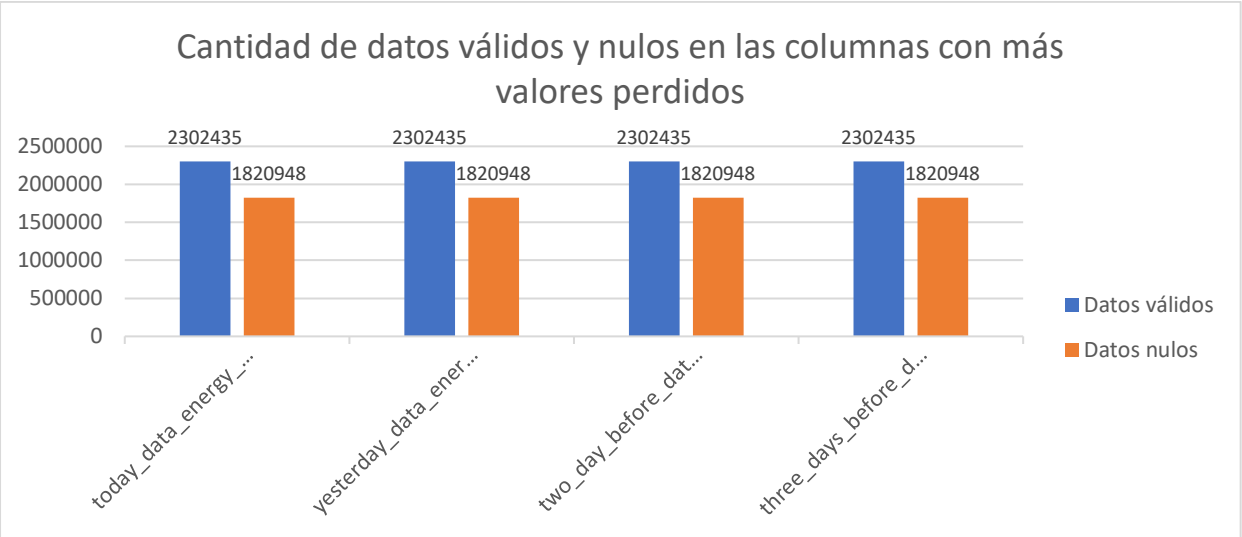


Figura 33**:** *Cantidad de datos válidos y nulos de las variables con más variables perdidos.*

Para visualizar la forma de los datos el programa generó histogramas para cada una de las variables; es decir, más de 70 histogramas. Adicionalmente se generaron gráficos tipo bloxplot y de dispersión en igual cantidad. Dada la cantidad de variables solo se muestra una de cada tipo para dos variables: una sin datos nulos (**Figura** *34*, **Figura** *36*, **Figura** *38*) y otra con datos nulos (**Figura** *35*, **Figura** *37*, **Figura** *39*).

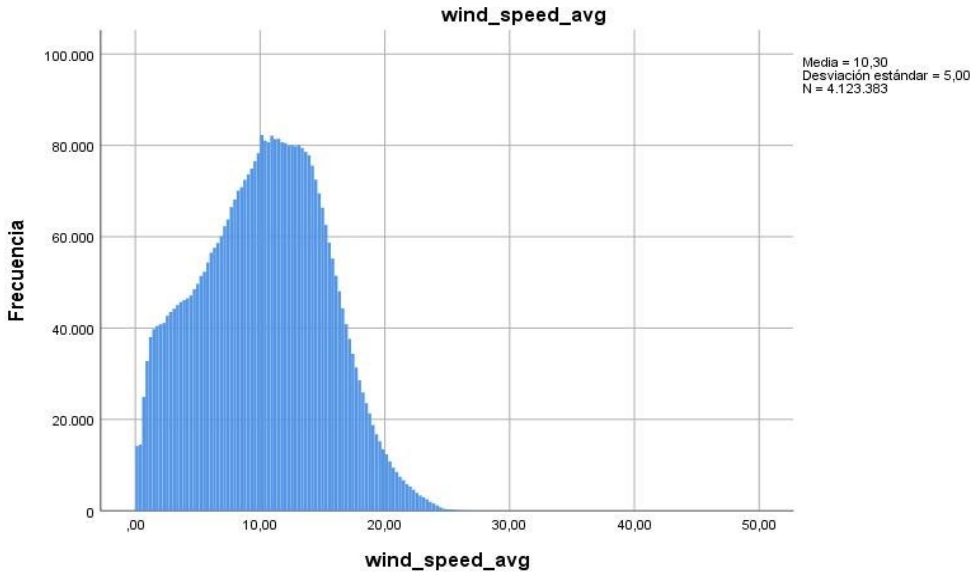


Figura 34**:** *Histograma de una variable sin datos nulos generado por el software SPSS.*

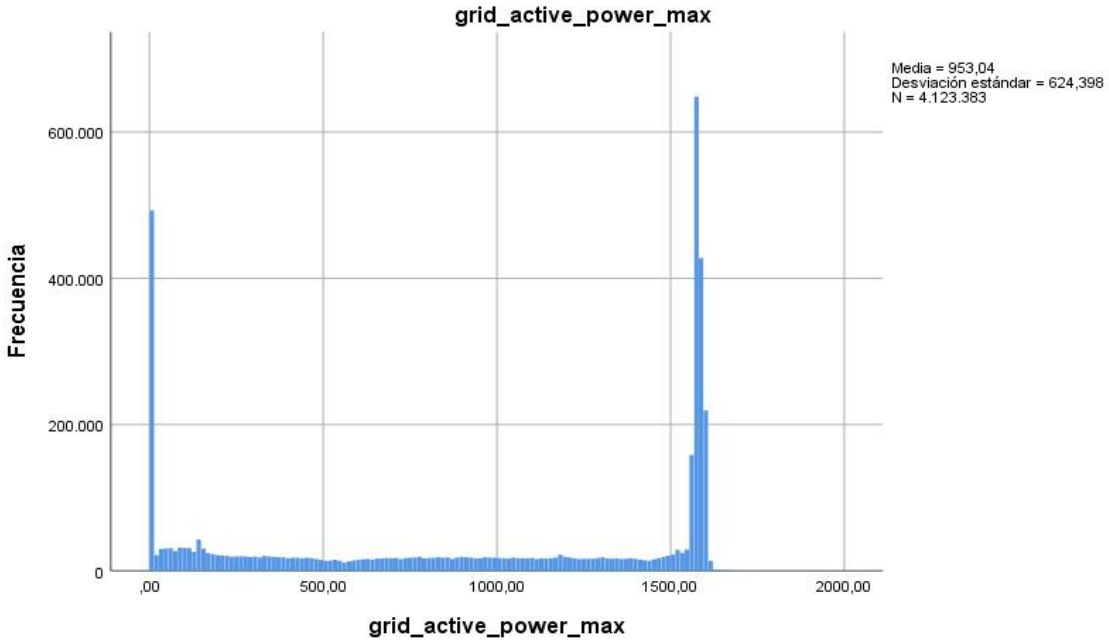


Figura 35**:** *Histograma de una variable con datos nulos generado por el software SPSS.*

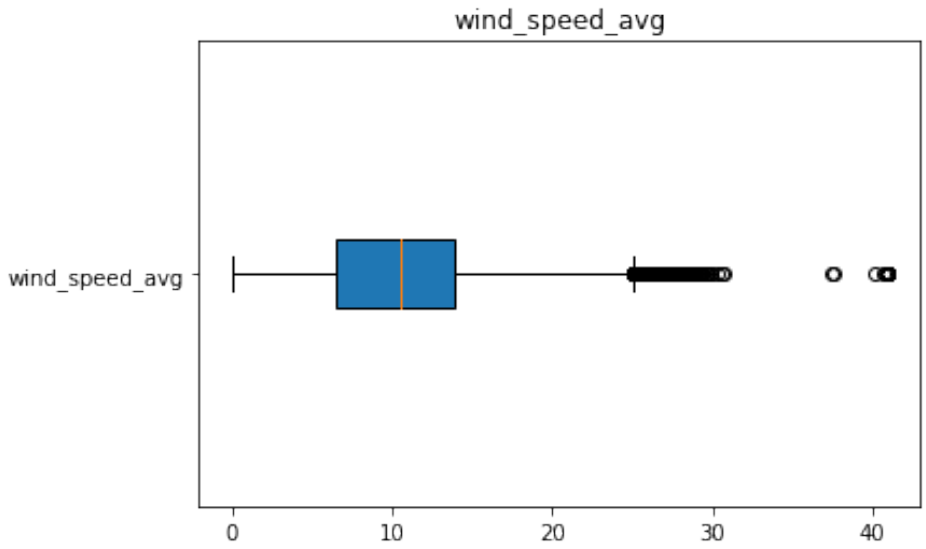


Figura 36**:** *Boxplot de una variable sin datos nulos generado por el software SPSS.*

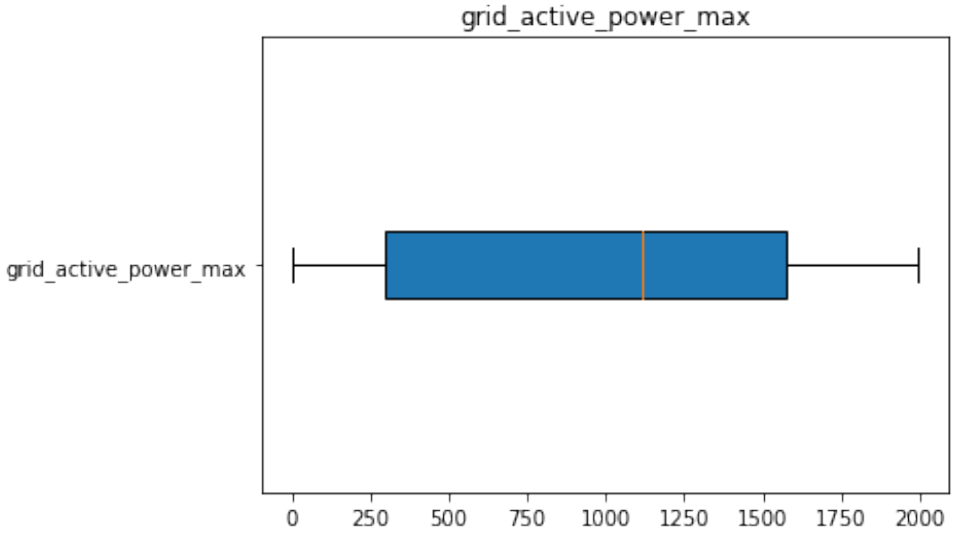


Figura 37**:** *Boxplot de una variable con datos nulos generado por el software SPSS.*

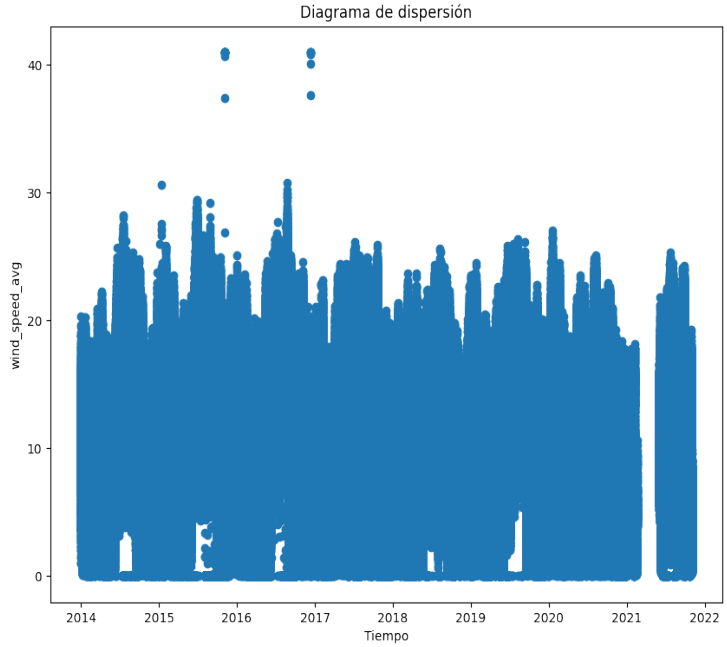


Figura 38**:** *Diagrama de dispersión de una variable sin datos nulos generado por el software SPSS.*

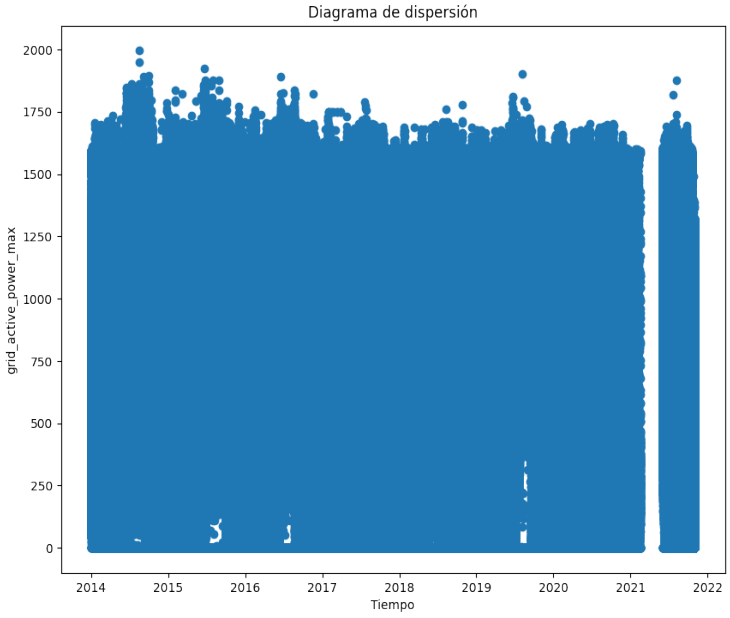


Figura 39**:** *Diagrama de dispersión de una variable con datos nulos generado por el software SPSS.*

Al fijarse en los diagramas de dispersión se puede notar que existe un lapso de tiempo en que la cantidad de datos es nula en ambos casos. Esto sería aceptable en la variable que contiene datos nulos, pero no en aquella que no los presentó. Por medio de la observación del diagrama de dispersión se consigue notar que los datos ausentes se presentan en el mismo lapso de tiempo. Esto lleva a pensar que en la base de datos existe un periodo de tiempo en el que no

se tomaron datos y no fueron incluidos en la base si no que fueron eliminados provocando que ese lapso de tiempo pase desapercibido.

6.1.2 Manejo de datos ausentes.

Al tener identificados los problemas que existen, se cargó la base de datos al programa Python teniendo en cuenta una serie de tareas necesarias para limpiarla:

- Eliminar las columnas cuyo porcentaje de datos ausentes es alto.
- Eliminar las filas que contienen datos ausentes.
- Trabajar con la base de datos sin ausencia de lapsos de tiempo. Si este lapso ausente se encuentra en medio de dos lapsos de tiempo que no presentan problemas, se escogerá aquel que tenga la mayor cantidad de datos y el resto será eliminado junto con el lapso ausente.

La primera tarea se lleva a cabo debido a que, si se opta por otra de las opciones recomendadas en la bibliografía tal como lo es, el relleno de datos con el valor de la moda o un valor medio, estos van a volverse demasiado recurrentes y no serán de gran ayuda al momento de entrenar los modelos de Machine Learning.

La segunda tarea se tomó considerando el tamaño de la base de datos en comparación con los valores ausentes. Luego de la eliminación de las columnas con un gran porcentaje de valores faltantes se calculó el porcentaje de datos nulos en las columnas restantes, dando como resultado un valor máximo de 0.001091%; es decir, una cantidad ínfima de datos.

Al ejercer la tercera tarea se comprobó que el lapso de tiempo faltante se da desde la fecha 2021-02-19 hasta el 2021-05-31. Dado que la base contempla datos desde el año 2014 se eliminaron todos los datos pertenecientes a una fecha posterior al 2021-02-19.

Una vez se hayan completado las tareas antes planteadas se tiene una base sin datos nulos. Esto se comprueba por medio del recálculo de datos nulos. Al dar un resultado del 0% se realiza nuevamente un escaneo de los datos por medio de estadística descriptiva en el software SPSS.

6.1.3 Estadística descriptiva de la base preprocesada.

Se realizó el mismo procedimiento dentro del software SPSS que para la base de datos sin procesar y se obtuvo una nueva tabla en la que constan las variables, cantidad de datos válidos, cantidad de datos nulos y valores estadísticos (**Tabla** 4). Así mismo se generaron nuevos histogramas, Boxplot y diagramas de dispersión que, al igual que en el caso pasado, no serán mostrados en su totalidad debido a la magnitud de los mismos.

Tabla 4**:** *Muestra de la estadística descriptiva de la base preprocesada.*

Variable	wind_speed_avg	wind_speed_max	converter_reactive_power_avg	converter_reactive_power_min	generator_speed_avg	ambient_temperature_max
Número de datos nulos	0	0	0	0	0	0
Media	10,2846	13,449	0,124	-20,2368	14,42	13,96
Mediana	10,49	13,69	0,00	-11,10	17,30	13,6
Moda	0,00	0,00	0,00	-3,41	18,50	13,1
Desviación	5,03	6,14	2,42	36,22	5,94	1,96
Varianza	25,307	37,71	5,88	1312,49	35,291	3,869
Rango	40,97	40,97	1776,60	1757,74	29,04	850,0
Mínimo	0,00	0,00	-1749,12	-1750,05	-9,99	0,0
Máximo	40,97	40,97	27,48	7,69	19,05	850,0

En la **Figura** *40* se puede observar la cantidad de datos válidos de la base preprocesada y, tal como se nota, es de un 100% mientras que en la **Figura** *41* se puede observar una comparación entre la cantidad de datos de la base original en comparación con la de la base preprocesada.

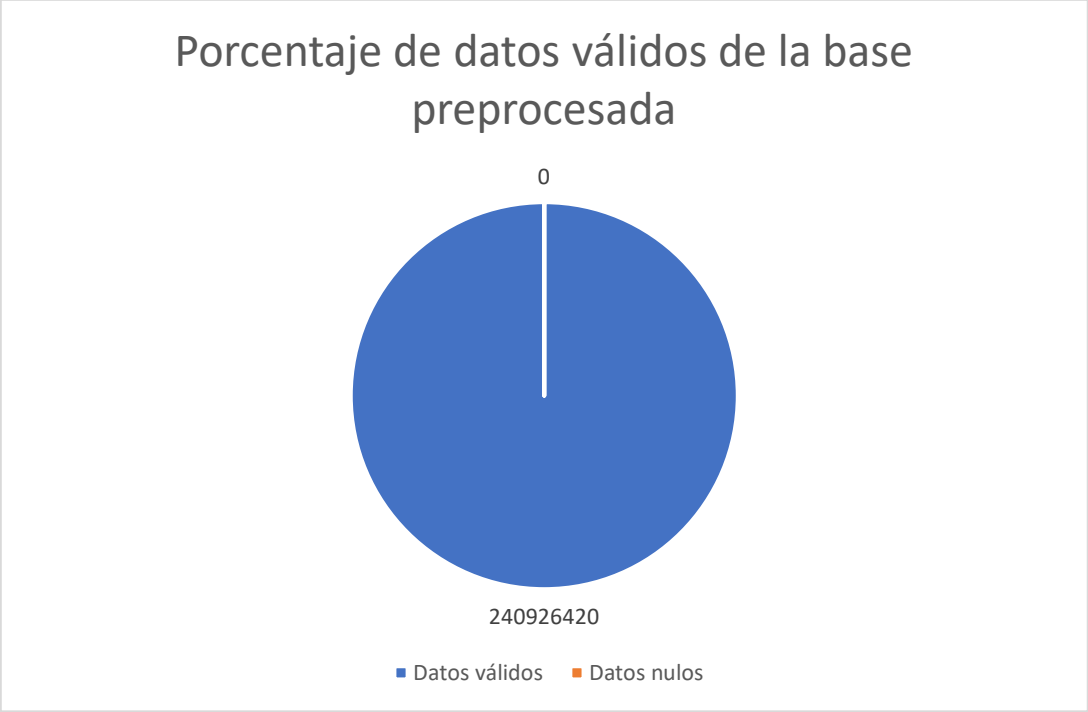


Figura 40**:** *Porcentaje de datos válidos de la base preprocesada.*

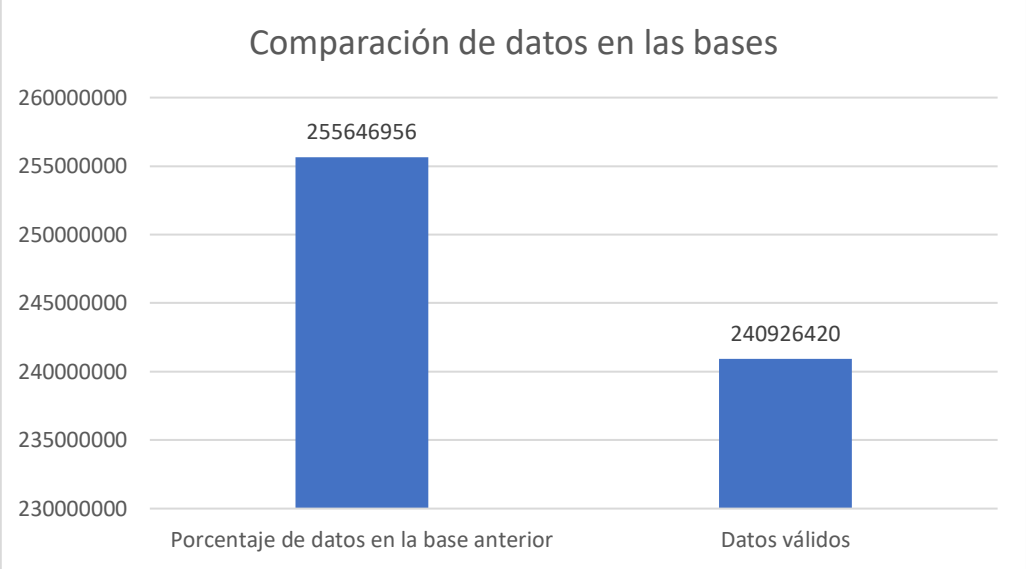


Figura 41**:** *Comparación sobre la cantidad de datos válidos entre bases.*

Por medio del estudio de la **Figura** *41* se nota que la base de datos sufrió un cambio en cuanto a cantidad de datos válidos. Esta diferencia es de 14720.536 datos, lo que representa el 5.75% de la base original. Esto también se ve reflejado en la **Tabla** 4.

Dicho esto, se procede a verificar si los gráficos generados sufrieron un cambio notorio porque esto significaría que los datos cambiaron sus propiedades. Para una mejor visualización se presentan los gráficos de las mismas variables presentadas para la base original.

Al comparar los histogramas de la **Figura** 34 y de la **Figura** *42*, se nota que, a pesar de eliminar algunos datos, los valores estadísticos tales como media y desviación estándar apenas variaron. Esto indica que la base mantiene sus propiedades y no se vio demasiado afectada por la limpieza realizada.

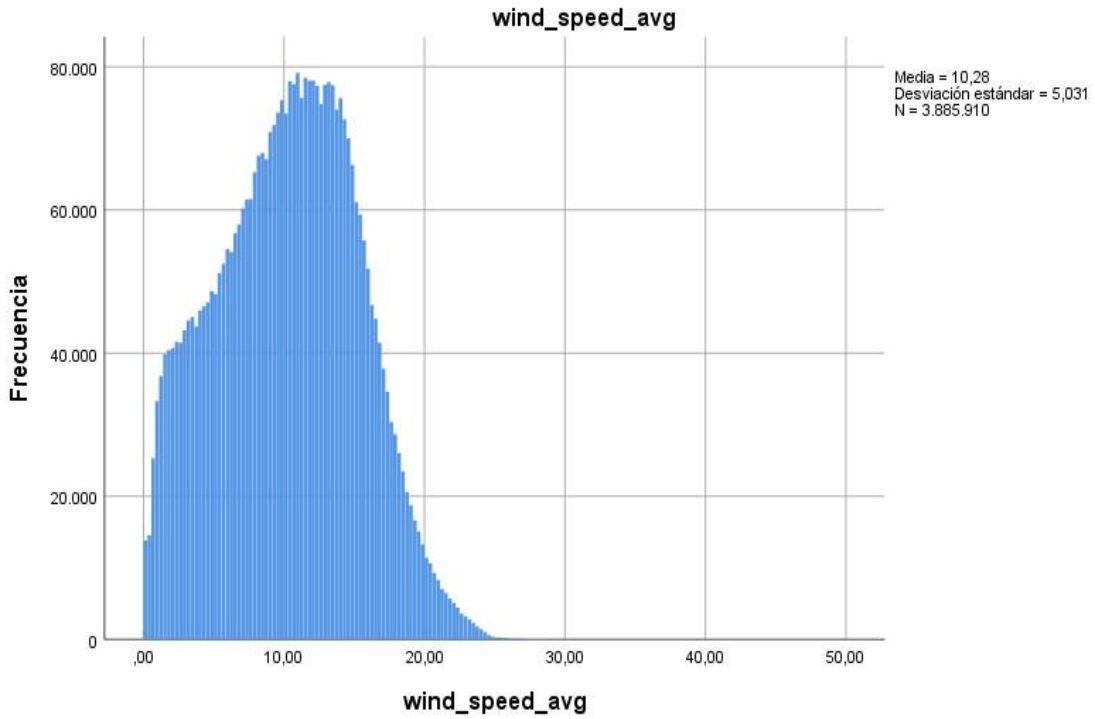


Figura 42**:** *Histograma de una variable de la base preprocesada generada por el software SPSS.*

Al comparar los Boxplot de **Figura** 36 con el Boxplot de la **Figura** *43* se verifica lo afirmado anteriormente. Esto se demuestra debido a que el valor de los datos atípicos mantiene la misma orientación que aquellos encontrados en la base original, es decir, la base sin procesar.

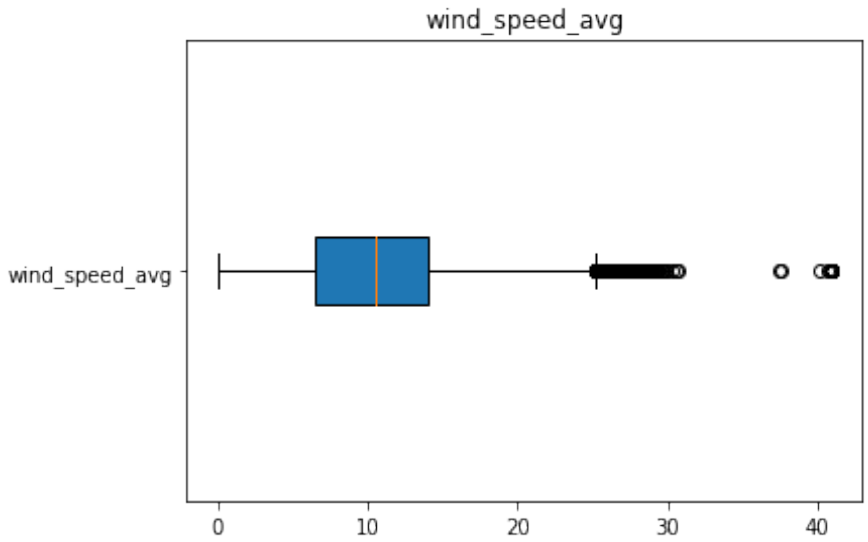


Figura 43**:** *Boxplot de una variable de la base preprocesada generada por el software SPSS.*

Finalmente, tal y como se observa en el diagrama de dispersión dado en la **Figura** *44*, ya no hay lapsos de tiempos desaparecidos que puedan generar incongruencias al momento de entrenar los algoritmos de machine learning.

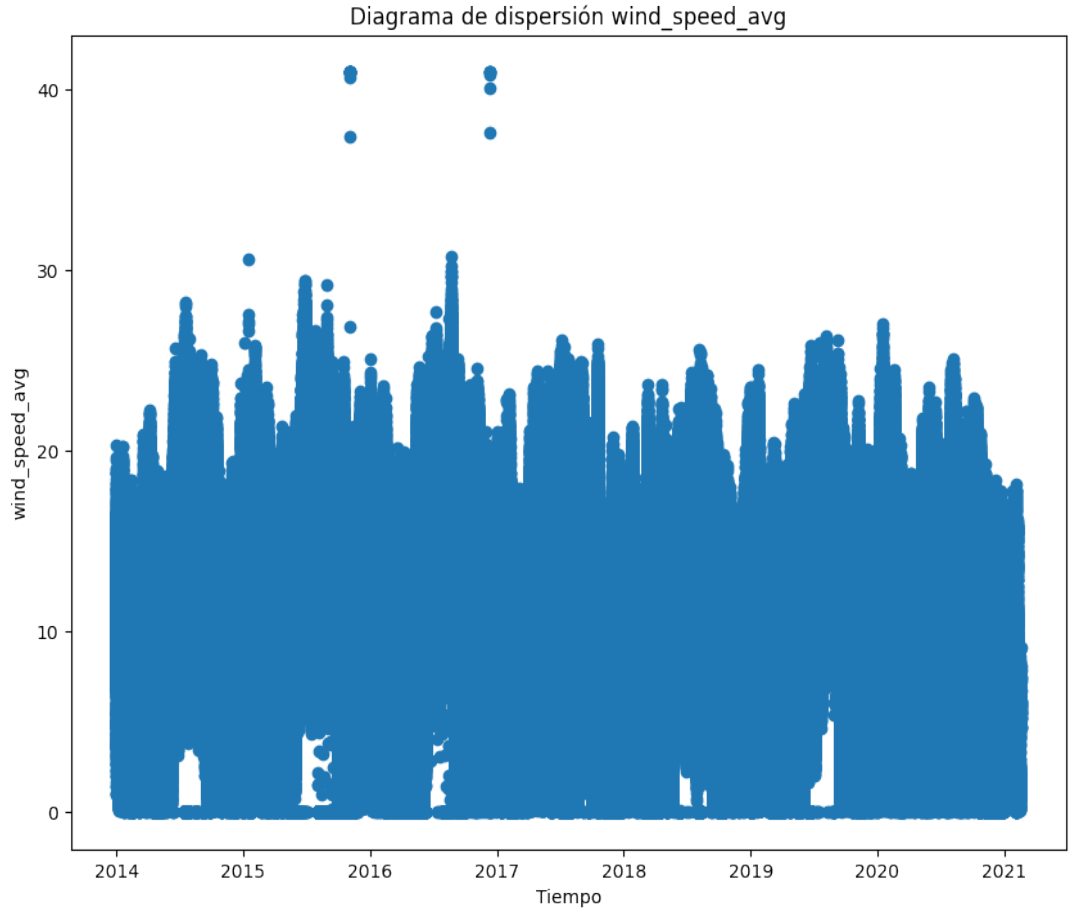


Figura 44**:** *Diagrama de dispersión de una variable de la base preprocesada generada por el software SPSS.*

6.1.4 Selección de variables.

Para la selección de variables, se usó el método R2. Luego de ingresar el código necesario se determinó que veintidós variables tienen una relación superior al 5% requerido. Estas variables se presentan en la **Tabla** 5.

Tabla 5: *Variables seleccionadas para el entrenamiento en machine learning.*

	Variables seleccionadas	R2
1	pitch_motor_temperature_2_max	0.163984
2	igbt_temperature_max	0.147230
3	pitch_motor_temperature_3_max	0.146020
4	pitch_motor_temperature_1_max	0.145978
5	rectifier_temperature_max	0.137181
6	ac_inductor_temperature_max	0.129842
7	wind_speed_avg	0.123232
8	pitch_converter_temperature_2_max	0.113548
9	step_up_igbt_temperature_max	0.113242
10	pitch_converter_temperature_1_max	0.112771
11	winding_temperature_max	0.111617
12	dc_link_capacitors_temperature_max	0.103234
13	dc_inductor_temperature_max	0.094346
14	grid_active_power_avg	0.091321
15	grid_I2_avg	0.090598
16	grid_I3_avg	0.090223
17	grid_I1_avg	0.089704
18	pitch_converter_temperature_3_max	0.088907
19	wind_speed_min	0.088499
20	grid_active_power_max	0.087785
21	wind_speed_max	0.086931
22	grid_active_power_min	0.059698

Una vez escogidas las variables se da por terminado el proceso de limpieza y preparación de la base de datos y se procede al entrenamiento de los algoritmos de machine learning.

6.2 Aplicación de los algoritmos de machine learning.

Antes de realizar el entrenamiento se debe tomar en consideración la cantidad masiva de datos que posee la base preprocesada. Al utilizar dicha base se tendrá una gran cantidad de datos que ralentizarán el entrenamiento de los algoritmos y es por este motivo que se realizó un balanceo de datos mediante el cual obtuvimos una muestra que cuenta con las 22 variables seleccionadas anteriormente y 5551 datos para cada una de estas. Esta muestra cuenta con una mayor cantidad de datos referentes a la falla del módulo IGBT garantizando que los algoritmos usados obtengan un buen porcentaje en las métricas de rendimiento a implementarse debido a su buen desempeño al detectar las fallas y no los momentos en que no existieron anomalías.

Para el análisis de los algoritmos de machine Learning se usan las métricas de precisión explicadas en el marco teórico. Dichas métricas fueron aplicadas en cada uno de los modelos.

6.2.1 Random Forest

Luego de realizar el test del algoritmo Random Forest se generó la matriz de confusión dada en la **Figura** *45* mientras en que la **Tabla** 6 se muestran los valores ordenados acorde la distribución mostrada en la **Figura** 8.

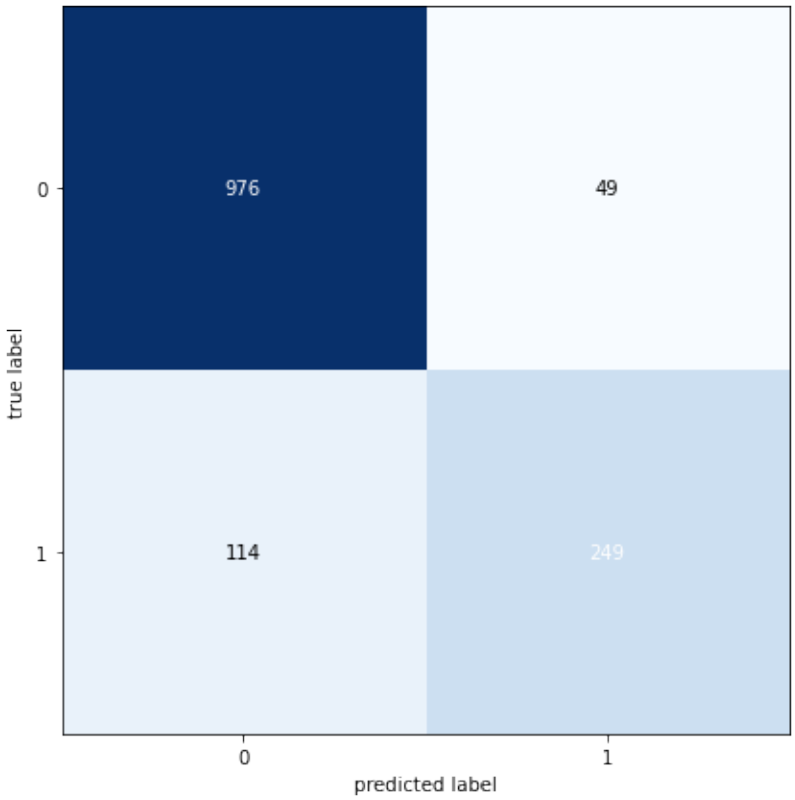


Figura 45**:** *Matriz de confusión para el algoritmo de Random Forest.*

Tabla 6**:** *Valores asociados a la matriz de confusión para Random Forest***.**

	0	**1**
0	976	49
1	114	249

En la **Tabla** 7 se encuentran los valores de las métricas de rendimiento aplicadas en el software Python con ayuda de la librería sklearn.

Tabla 7**:** *Valores arrojados por las métricas de rendimiento para el algoritmo de Random Forest.*

	0	**1**
Precisión	0,9	0,84
Recall	0,95	0,69
F1-score	0,92	0,75

Support	1025	363
Acurracy promedio	0,8825	

6.2.2 SVM

Luego de realizar el test del algoritmo SVM se generó la matriz de confusión dada en la **Figura** *46* y sus valores se dan en la **Tabla** 8, mientras en que la se muestran los valores ordenados acorde la distribución mostrada en la **Figura** 8. Por otra parte, en la **Tabla** 9 se muestran los valores de las métricas de presición calculadas para este algoritmo.

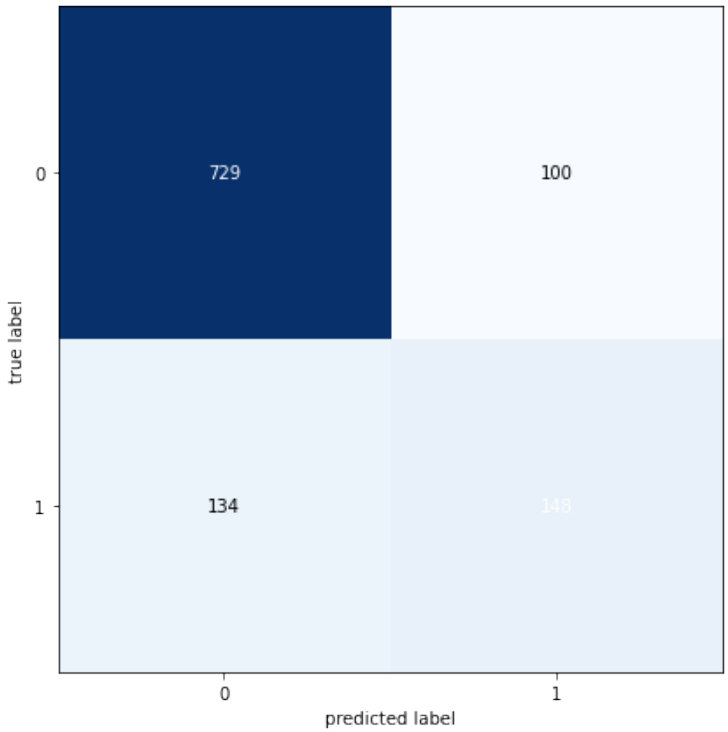


Figura 46**:** *Matriz de confusión para el algoritmo de SVM*

Tabla 8**:** *Valores asociados a la matriz de confusión para SVM*

	0	**1**
0	729	100
1	134	148

Tabla 9**:** *Valores arrojados por las métricas de rendimiento para el algoritmo de SVM*

	0	**1**
Precisión	0,84	0,6
Recall	0,88	0,52
F1-score	0,86	0,56

Support	829	282
Acurracy promedio	0,8825	

6.2.3 KNN

Dado que uno de los parámetros requeridos en este algoritmo es la cantidad de vecinos a usar, se determinó primero este número por un medio gráfico (**Figura** *47*) al comparar el Acurracy obtenido por medio del uso de varios KNN.

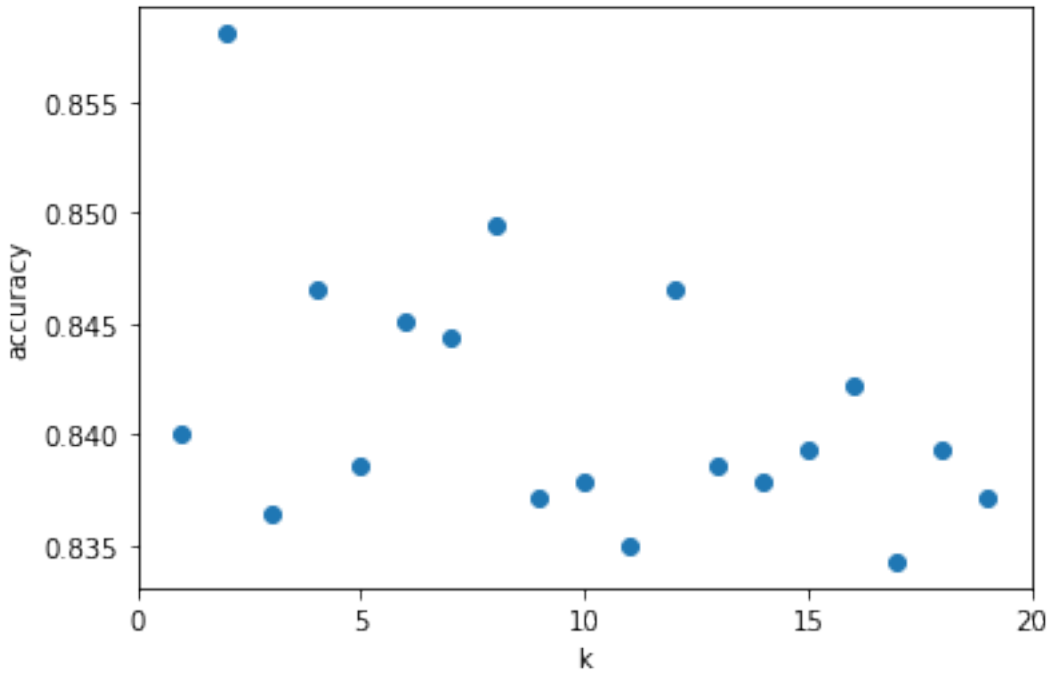


Figura 47**:** *Valores de Acurracy para diferentes valores de KNN.*

Se observa que el valor de Acurracy más elevado se encuentra en el valor 2; sin embargo, obtener un valor muy cercano a 1 indica que el modelo está sobreentrenado, se optó por usar el valor de 8, cuyo valor de Acurracy se acopla más a las exigencias del modelo.

Una vez se realice el entrenamiento con los parámetros adecuados, se obtiene la matriz de confusión mostrada en la **Figura** *48*, sus valores en la **Tabla** 10 y los resultados de las métricas de precisión en la **Tabla** 11.

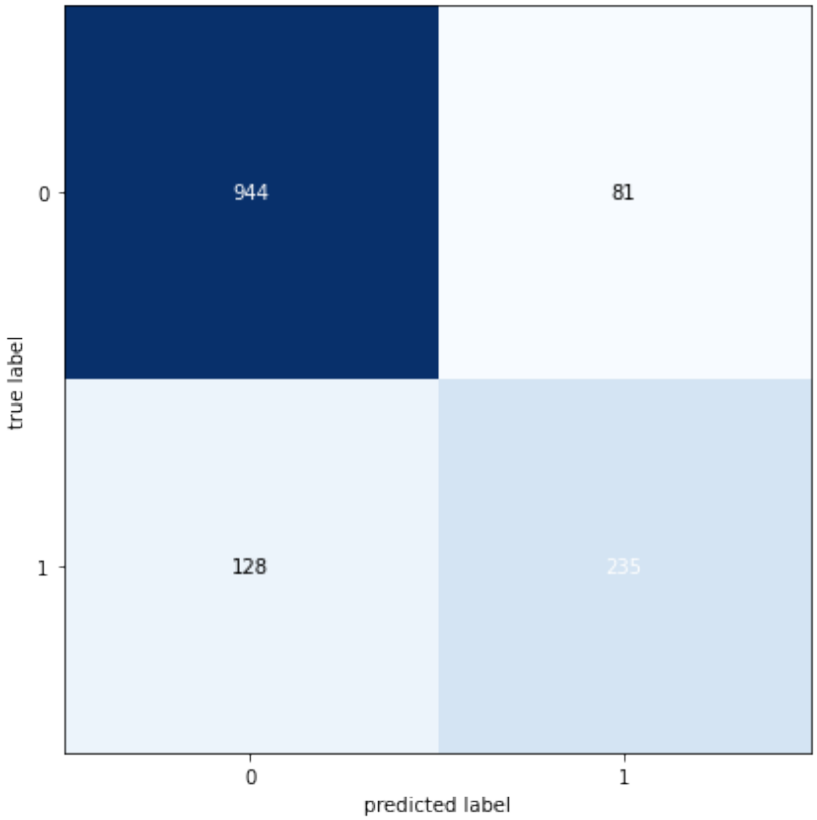


Figura 48**:** *Matriz de confusión para el algoritmo KNN*

Tabla 10**:** *Valores asociados a la matriz de confusión para KNN.*

	0	**1**
0	944	81
1	128	235

Tabla 11**:** *Valores arrojados por las métricas de rendimiento para el algoritmo KNN.*

	0	**1**
Precisión	0,88	0,74
Recall	0,92	0,65
F1-score	0,9	0,69
Support	1025	363
Acurracy	0,85	
promedio		

6.2.4 Naive Bayes

Al igual que en los algoritmos anteriores, luego de realizar el test del algoritmo Naive Bayes se generó la matriz de confusión dada en la **Figura** *47*, los valores asociados a esta se

muestran en la **Tabla** 12 mientras que en la **Tabla** 13 se muestran los resultados de las métricas obtenidas para este algoritmo.

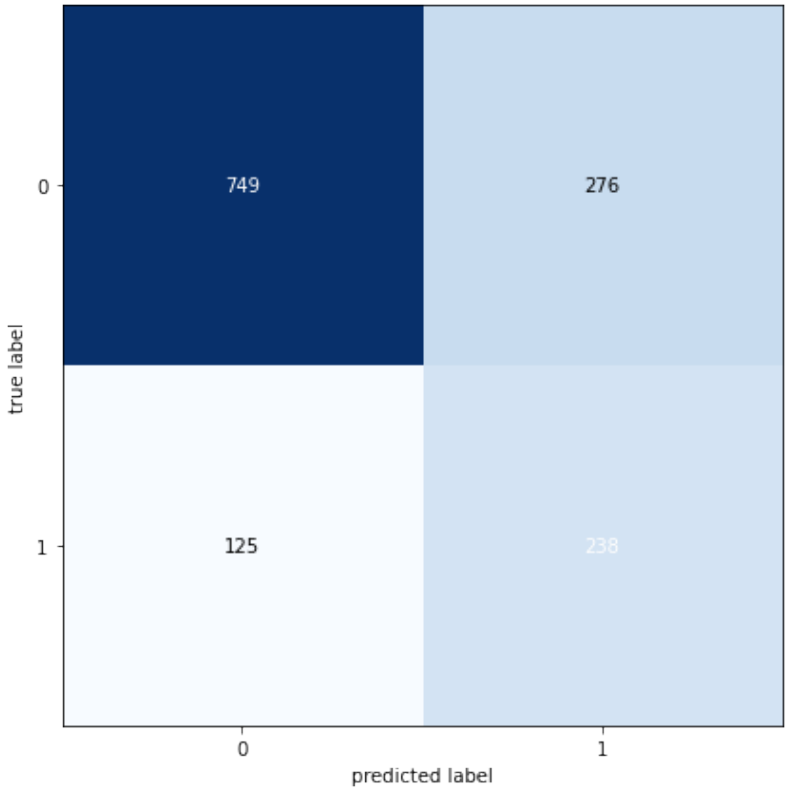


Figura 49**:** *Matriz de confusión para el algoritmo de Naive Bayes*

Tabla 12**:** *Valores asociados a la matriz de confusión para Naive Bayes.*

	0	**1**
0	749	200
1	125	238

Tabla 13**:** *Valores arrojados por las métricas de rendimiento para el algoritmo Naive Bayes.*

	0	**1**
Precisión	0,86	0,46
Recall	0,73	0,66
F1-score	0,79	0,54
Support	1025	363
Acurracy promedio	0,7110	

Dado que el trabajo se concentra en la detección de fallas se toma en cuenta los valores ubicados en el valor 1. Teniendo en cuenta esta observación y por medio del software Excel se

creó la **Figura** *50* en donde constan los valores de las métricas al predecir fallas por medio de cada uno de los algoritmos usados.

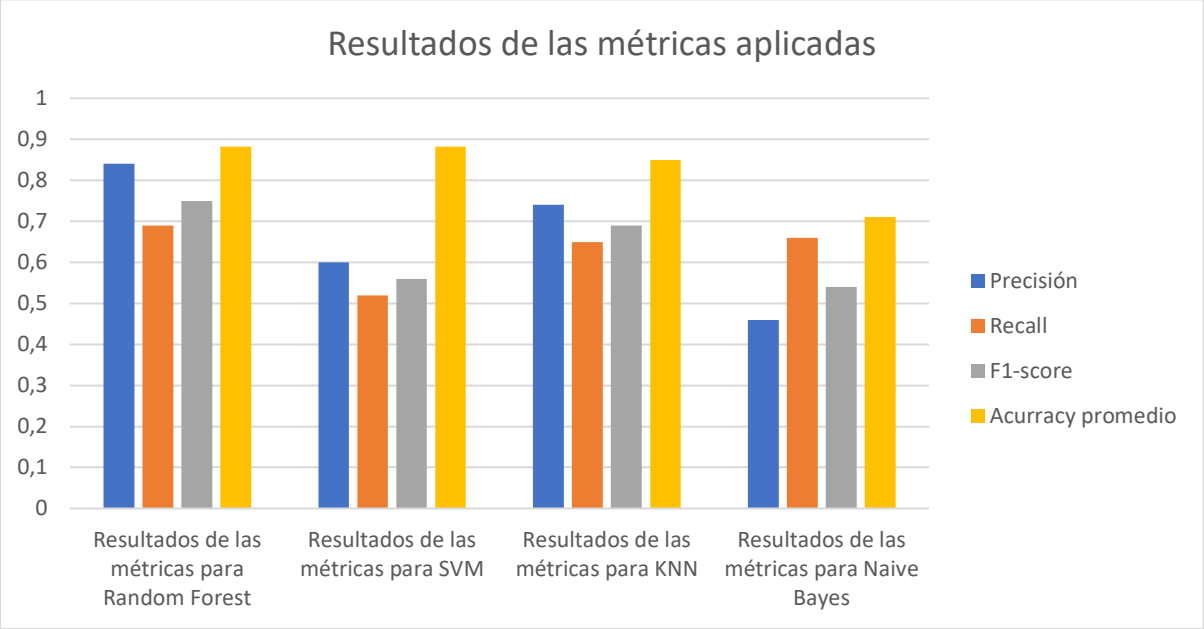


Figura 50**:** *Comparación de los valores resultantes de las métricas aplicadas en cada uno de los algoritmos de machine learning*

6.3 Curvas ROC

Las curvas ROC se basan en la magnitud de la métrica AUC por lo que mientras más aproximado a 1 sea dicho valor, mejor desempeño habrá tenido el algoritmo de machine learning utilizado.

En la **Tabla** 14 se pueden encontrar los diversos valores de la métrica AUC para cada uno de los algoritmos de machine learning implementados.

Tabla 14**:** *Valores de la métrica AUC para los diferentes valores.*

Modelo	Random Forest	SVM	KNN	Naive Bayes
Valor de AUC	0,905	0,76	0,87	0,76

Con dichos valores se generó una curva ROC para cada uno de los algoritmos usados. Esto se hace para la mejor apreciación individual de cada una de ellas.

Las curvas ROC dadas por el software (**Figura** *51*, **Figura** *52*, **Figura** *53*, **Figura** ***54***) se presentan a continuación.

6.3.1 *Curva ROC para algoritmo de Random Forest*

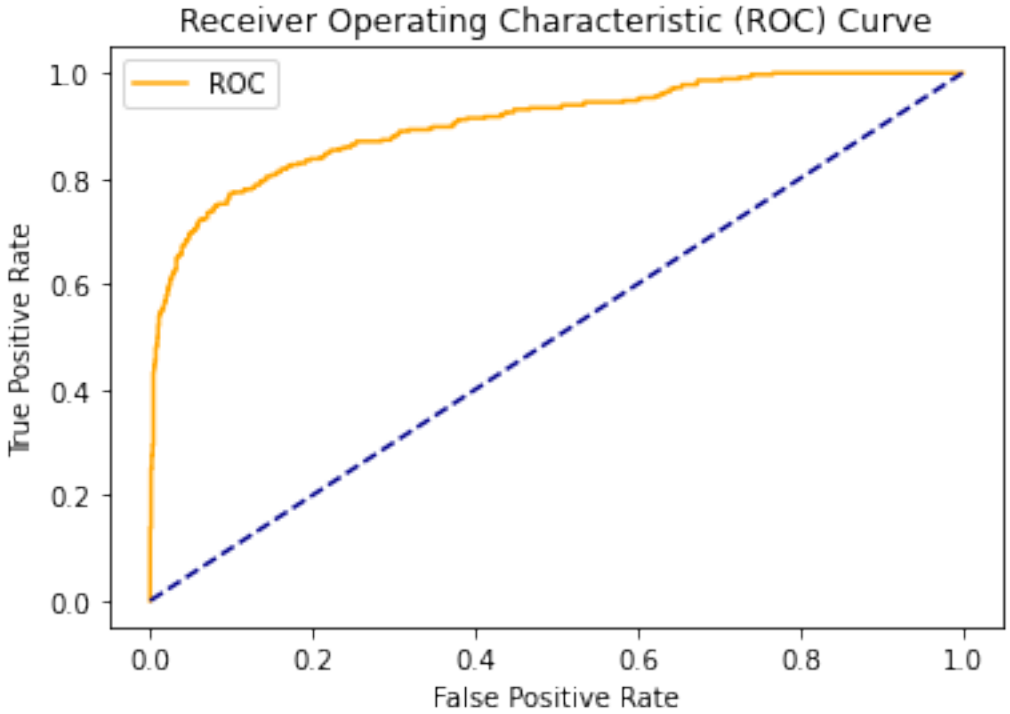


Figura 51**:** *Curva ROC para el algoritmo de Random Forest.*

6.3.2 *Curva ROC para algoritmo de Vecinos cercanos (KNN).*

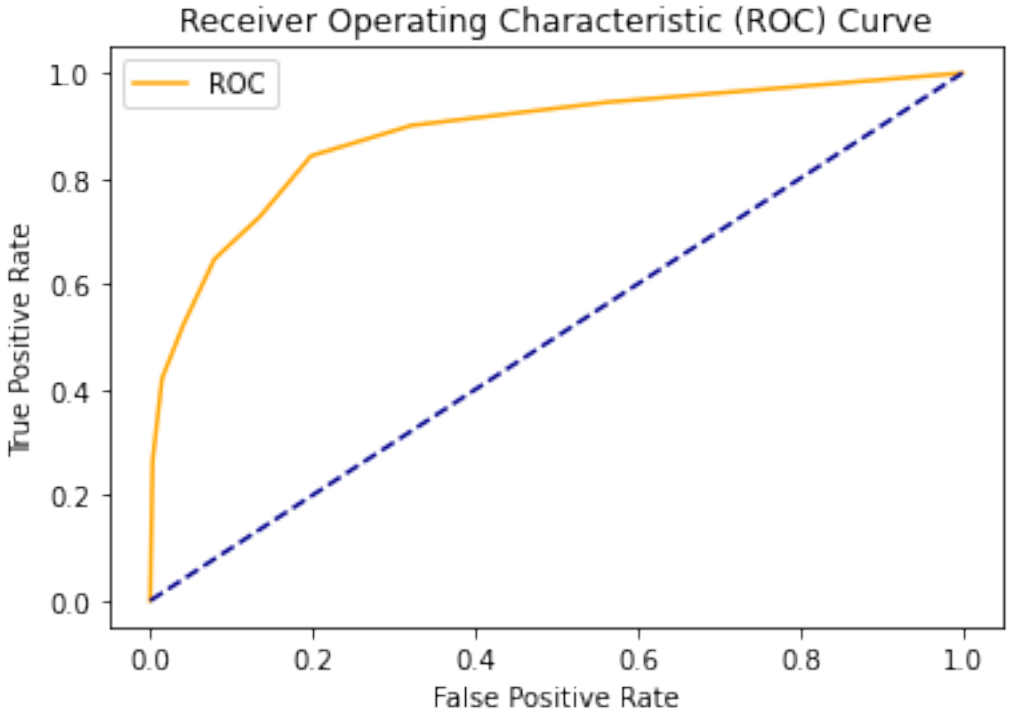


Figura 52**:** *Curva ROC para el algoritmo de KNN.*

6.3.3 Curva ROC para algoritmo de SVM.

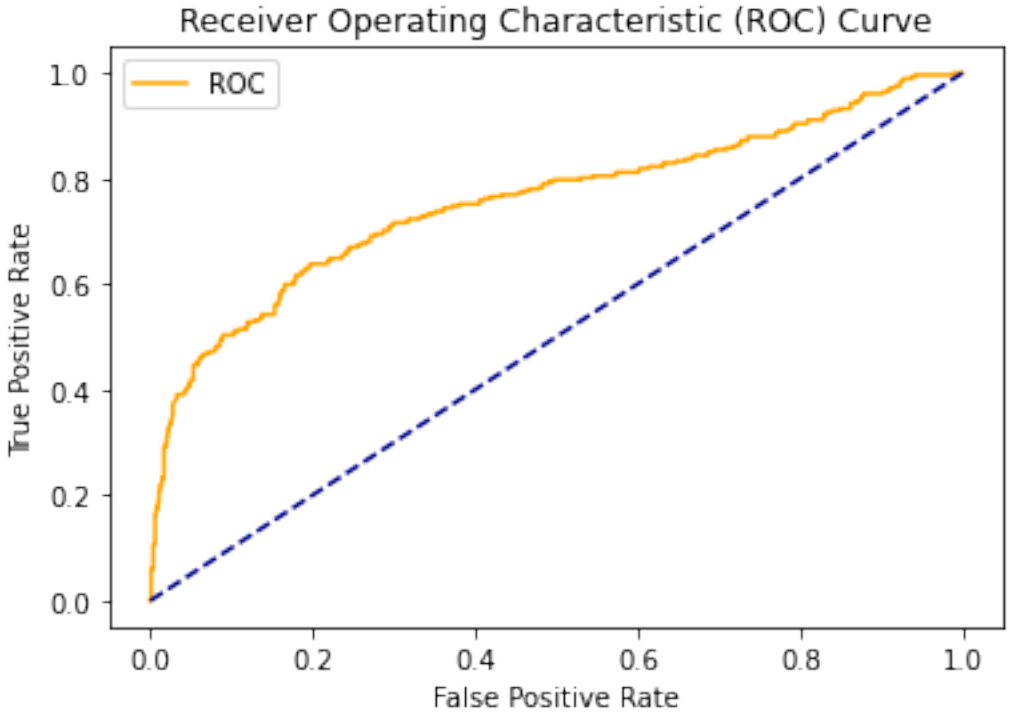


Figura 53**:** *Curva ROC para el algoritmo SVM.*

6.3.4 Curva ROC para algoritmo de Naive Bayes.

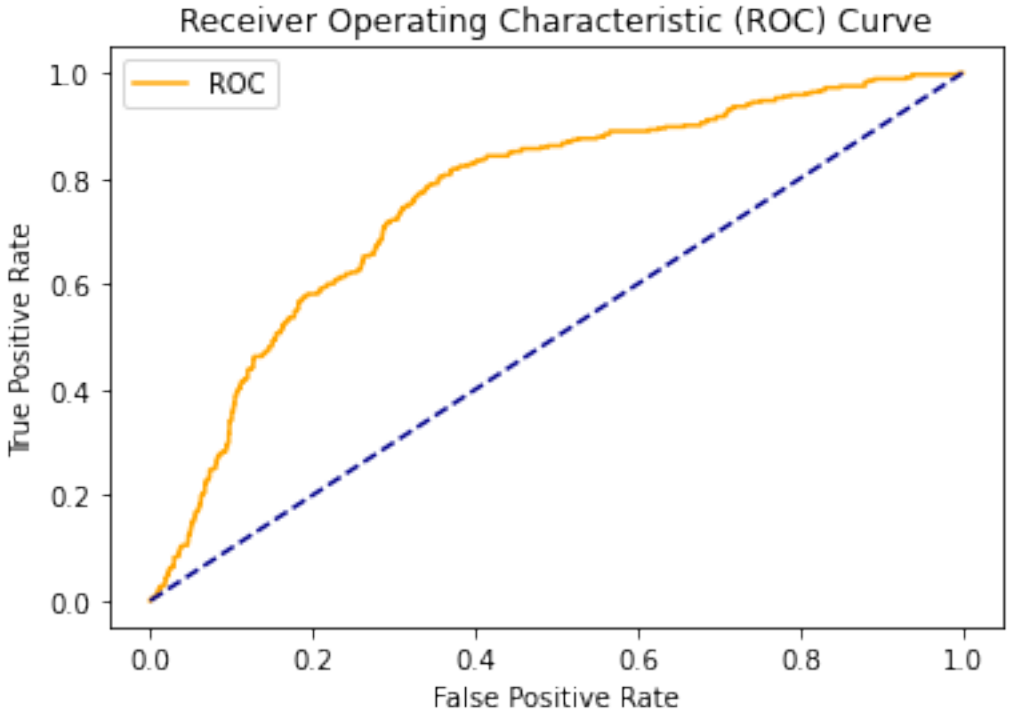


Figura 54: *Curva ROC para el algoritmo Naive Bayes.*

6.3.5 Comparación entre las curvas ROC.

Desde la **Figura** 49 hasta la Figura 52 se observan las curvas ROC para los algoritmos de machine Learning. Estas muestran su eficacia al momento de predecir las fallas del módulo IGBT, pero por separado no reflejan la ventaja o desventaja de cada algoritmo sobre otro.

Para una mejor apreciación entre las diferencias, se generó una curva extra en la que se encuentran superpuestas entre si todas las curvas ROC de los algoritmos usados. Esta puede ser visualizada en la **Figura** *55*.

Dicha curva permite identificar gráficamente al algoritmo con mejor desempeño en el trabajo realizado puesto que se nota fácilmente cual es la curva dominante. En dicha figura también se hace constar el valor de AUC que, de acuerdo a su magnitud, refleja en desempeño que tuvo el algoritmo al predecir las fallas.

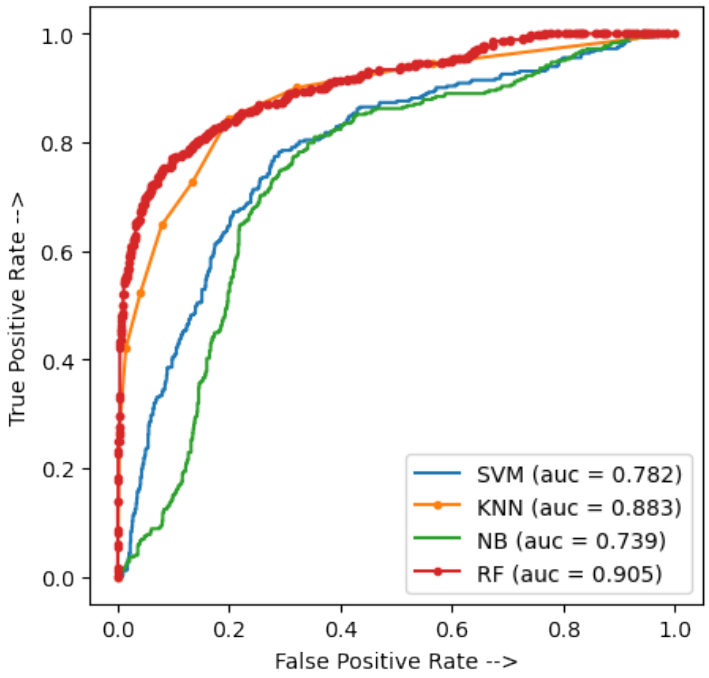


Figura 55: *Conjunto de curvas ROC generadas.*

Se nota claramente que la curva perteneciente al algoritmo Random Forest domina sobre las demás. Esto se comprueba con su valor de 0.905 en la métrica AUC y el análisis de la **Tabla** 6 en donde se aprecia que los valores de verdaderos positivos y falsos positivos son superiores a sus contrapartes. Por otra parte, al fijarse en los valores de las métricas para el algoritmo Random Forest (**Tabla** 7) y compararlos con los valores de las mismas métricas para los demás algoritmos, se demuestra que los datos para este algoritmo son los que dominan.

El algoritmo KNN se queda atrás por muy poco, de hecho, en la **Figura** 55 se puede observar cómo su curva ROC choca en muchos puntos con la curva ROC perteneciente a Random Forest, luego se separa momentáneamente y vuelve a tener la misma tendencia. Sus valores respecto a métricas afirman lo mismo puesto que son valores muy buenos.

Las curvas ROC pertenecientes a los algoritmos SVM y Naive Bayes, únicamente se encuentran con las curvas de las dos métricas anteriores al inicio y al final de su trayectoria. Sus valores en las métricas de rendimiento, si bien no son pésimos, no tiene comparación con los obtenidos mediante los algoritmos de Random Forest y KNN. Esto puede deberse a la propia naturaleza de los algoritmos o a la falta de ajustes en los hiperparámetros adicionales que ofrece cada una de ellas.

La curva ROC de la **Figura** 55 demuestra que el algoritmo Random Forest es el que mejor desempeño tuvo, seguido por los algoritmos KNN, SVM y NV respectivamente; sin embargo, cabe recalcar que ninguna de las trayectorias de las curvas de los algoritmos usados se ubica por debajo de la línea guía o, lo que es lo mismo, tienen un valor de AUC inferior a 0.5, todos los algoritmos clasifican de manera satisfactoria la variable falla.

7. Discusión

Luego del preprocesamiento de la base de datos se perdió el 5.75% de datos válidos sin afectar las características iniciales de la base y, por medio del método R2 para la selección de variables, se escogieron aquellas con más relación a la variable de falla. Esto dio como resultado el uso de veintidós variables cuyos valores son valiosos para el entrenamiento. El éxito de esta selección se ve reflejada en la eficiencia de los algoritmos aplicados al momento de predecir las fallas debido a que cada investigador, a pesar de seguir una metodología similar, toma decisiones basándose en las necesidades de cada caso. El proyecto de investigación de Francisco y Rodrigo (2022) cuenta con un objetivo basado en el preprocesamiento de una base de datos. La metodología usada coincide en su gran mayoría con los usados en este trabajo. Las partes que no concuerdan son aquellas referentes a la normalización de los datos que consiste en una técnica de preprocesamiento de datos que se utiliza para ajustar las características o atributos de los datos dentro de un rango específico y mejorar su interpretación. Esto se debe a que en dicho trabajo se implementó una red LSTM, la cual obligatoriamente necesita este procedimiento de normalización para un buen funcionamiento. Dado que en este trabajo se usaron algoritmos en los que no es indispensable tener datos normalizados, no fue necesario la implementación de esta parte. Por otra parte, la similitud entre los procedimientos usados en el presente trabajo y los de otros autores, además del hecho de que la base preprocesada mantenga características estadísticas similares a la original, muestra que el objetivo se cumplió satisfactoriamente.

Los valores de precisión obtenidos en el presente proyecto para los algoritmos Random forest, KNN, SVM y Naive Bayes son 88.25%, 85%, 88.25% y 71.1% respectivamente. Dado que estos valores son superiores al 50% se afirma que tienen la capacidad de detectar fallas. En cuanto a los modelos de clasificación, en el trabajo de Labañino Urbina (2019) se implementó el algoritmo Random Forest para la predicción de fallas en un sistema computacional. Si bien la naturaleza del fenómeno de estudio difiere mucho entre ambas investigaciones, el algoritmo trabaja de forma similar en ambos casos. Al momento de ser aplicado el algoritmo referido, y contando con tres árboles de decisión, obtuvo una precisión del 96.15% mismo que no está alejado del 88.25% obtenido en el presente proyecto.

Hay que recalcar que en el presente proyecto se usó una base de 22 variables con cinco mil quinientos cincuenta datos en cada una de ellas, el software Python y 19 árboles de decisión, en el trabajo de Labañino (2019) se desconoce la cantidad de variables que intervienen en el entrenamiento, pero se sabe que cada una de ellas contaba con 89 datos, se usó el software Java con la herramienta APACHE SPARK y tres árboles de decisión. De acuerdo a la página

Aprende Machine Learning (2022), una de las desventajas que posee este algoritmo es que no trabaja bien con bases de datos pequeñas, pero es muy bueno con bases de datos grandes sin importar que sus hiperparámetros no sean ajustados. Lo anteriormente dicho lleva a discernir que es probable mejorar los resultados del proyecto por medio del ajuste adecuado de los hiperparámetros.

Otros métodos usados por autores son las redes neuronales. Se tiene que Xie Y y Zao (2021), al usar redes CNN y LSTM en la predicción de fallas, obtuvieron valores de precisión del 68.55% y 76.6% respectivamente. Al igual que todos los algoritmos citados, clasifican de manera correcta a pesar de tener valores inferiores a los obtenidos en el presente trabajo.

La naturaleza de las fallas podría ser explicadas por medio de la selección de variables realizada puesto que el método usado mide la relación de cada una de estas con la alarma de falla del módulo IGBT de los aerogeneradores. Se tiene que las variables que más relación tienen con la alarma de falla son aquellas relacionadas principalmente con la temperatura del módulo, seguida por la velocidad del viento y la potencia. La mayor cantidad de variables seleccionadas están relacionadas con temperaturas por lo que se puede intuir que un gran porcentaje de estas fallas vienen dadas por problemas de sobrecalentamiento. La velocidad del viento es otro factor importante puesto que indica que a determinadas velocidades se corre el riesgo de que los componentes eléctricos, tales como el módulo IGBT, se quemen puesto que empiezan a trabajar a una mayor capacidad de la recomendada.

En cuanto a algoritmos supervisados, el trabajo de Cardenas, Vidal y Pozo (2021) usa un conjunto de ellos para la detección de fallas en turbinas eólicas y obtiene un valor de precisión del 98.65% con uso de KNN y 99.15% con SVM. Valencia Ordoñez (2021) en su trabajo para la detección de fallas en una subestación eléctrica alcanza valores de 87% al usar KNN y 99.21% con el algoritmo SVM, mientras que Dhanraj (2017), al usar el algoritmo NB en la predicción de fallas en turbinas eólicas. obtuvo una precisión del 85.67%. Esto da a entender que, al igual que en los trabajos citados, los algoritmos usados cumplen de manera correcta su trabajo a pesar de variar su eficiencia respecto a lo citado.

Las curvas ROC se basan en el valor de la métrica AUC obtenida, las cuales son de 0.905 para Random Forest (RF), 0.883 para el algoritmo de vecinos cercanos (KNN), 0.782 en Maquinas de soporte Vectorial (SVM) y 0.739 en el algoritmo Naive Bayes (NB). Estos valores se ven graficados en la Figura 53 donde se nota claramente que la curva correspondiente al algoritmo de Random Forest es el que mejor desempeño tiene mientras que Naive Bayes es aquel con el peor rendimiento. Esta afirmación se puede corroborar por medio de las demás métricas aplicadas puesto que los resultados de estas muestran al algoritmo Random Forest

como el que mejor confiabilidad tiene al momento de predecir las fallas en el módulo IGBT. En el proyecto de Navas (2021) se usaron curvas ROC para determinar el algoritmo de mejor desempeño. En dicho trabajo se usaron cuatro algoritmos: GXBoost, Random Forest, máquinas de soporte vectorial (SVM) y redes neuronales. Se hicieron tres entrenamientos para una variable de salida distinta y en dos de ellas el más confiable fue el algoritmo de Random Forest con un 0.71 de valor en la métrica AUC para el segundo problema y valores superiores a 0.5 en el tercero. Dicho esto, se determina que al igual en el trabajo mencionado, el algoritmo de Random Forest es aquel que mejor se desempeña al momento de predecir fallas y esto puede ser comprobado tanto numérica como gráficamente por medio del uso de los valores de la métrica AUC obtenidas o las curvas ROC generadas.

8. Conclusiones

Los pasos a seguir dados por la metodología CRISP-DM para el preprocesado de la base de datos muestran su validez puesto que son similares a los usados en otros trabajos de investigación en lo que se haya requerido este tratamiento de datos. Por otra parte, los datos estadísticos de la base original y de la preprocesada, al no distanciarse de forma abrupta, muestran que las características de ambas son similares por lo que su naturaleza no se vio afectada en demasía. Con lo anteriormente dicho y, teniendo en cuenta el porcentaje de efectividad que presentaron los algoritmos entrenados, se concluye que el preprocesamiento de la base de datos fue satisfactorio.

La selección de variables a usar en el entrenamiento de los algoritmos de Machine Learning muestra una clara predominancia a aquellas relacionadas con temperatura, velocidad del viento y potencia dando un mayor énfasis a la primera. Esto nos muestra claramente a aquellos factores que intervienen directamente en la falla del módulo IGBT por lo que, de acuerdo a los valores dados por el método R2, se averiguó que la temperatura tiene mucha más relación con las fallas que cualquier otra variable; es decir, la mayoría de fallas del módulo IGBT se producen por un sobrecalentamiento del sistema.

La calidad de los algoritmos usados y la capacidad que tienen para detección fallas en el módulo IGBT de los aerogeneradores de la central eólica Villonaco se mide respecto a los valores de las métricas de rendimiento usados luego del entrenamiento de cada uno de ellos. Dichos valores al estar siempre por encima del 50%, a excepción de la precisión individual para 1 en el algoritmo Naive Bayes, muestran que los algoritmos de Random Forest (RF), máquinas de vector soporte (SVM), vecinos cercanos (KNN) y Naive Bayes (NB) son capaces de predecir de forma eficiente dichas fallas; sin embargo, los resultados en las demás métricas nunca son inferiores al 50%, ya sea individualmente o promedio. Por lo anteriormente mencionado se concluye que todos los algoritmos son capaces clasificar los datos de fallas pero que su eficiencia varía debido a la naturaleza de cada uno de ellos.

Los valores de la métrica AUC muestran que el algoritmo de Random Forest ocupa el primer lugar en eficiencia al momento de clasificar las fallas. En segundo lugar, se encuentra el algoritmo de vecinos cercanos (KNN), como tercero se encuentra el algoritmo de máquinas de soporte vectorial (SVM) y finalmente el algoritmo de Naive Bayes (NB). La superioridad del algoritmo Random Forest y la secuencia mencionada anteriormente se pudo observar con mayor claridad por medio de la generación de curvas ROC que muestran dichos resultados gráficamente permitiendo emitir un juicio sin la necesidad de conocer los demás valores de las métricas usadas.

9. Recomendaciones

Al realizar el análisis estadístico de la base de datos inicial y de la base de datos preprocesada se pueden explorar opciones alternas al software SPSS. Esto se debe a que, si bien el programa cuenta con muchas herramientas estadísticas y de generación de imágenes, tiende a demorar en exceso cuando se trabaja con grandes bases de datos. Dado que los archivos generados por los sistemas SCADA durante largos periodos de tiempo tienen a tener grandes volúmenes de datos y gran tamaño, se pueden explorar opciones alternas para realizar dicho proceso.

Cada uno de los algoritmos posee sus propios parámetros los cuales pueden ser ajustados; sin embargo, al desconocer la función de cada uno de ellos es preferible trabajar con aquellos que vienen por defecto. En caso de querer ajustar alguno de ellos es necesario investigar el beneficio que conlleva modificarlos puesto que no todos tienen la misma influencia al momento de entrenar el logaritmo.

La gran mayoría de autores no mencionan los valores de las métricas de rendimiento de forma detallada puesto que únicamente se basan en los valores promedios de cada uno de estos. Por este motivo, al momento de realizar comparaciones entre los resultados de los valores obtenidos en las métricas de rendimiento y la de demás autores hay que tener en consideración estos factores.

Uno de los temas sugeridos para futuros trabajos es la optimización de uno de los 4 algoritmos usados en el presente trabajo de investigación puesto que, tal y como se mencionó anteriormente, existe la posibilidad de lograr incrementar su valor de precisión al momento de predecir las fallas de los módulos IGBT. Otro tema tentativo de investigación es el uso de la metodología planteada en este trabajo para la predicción de fallas en otros componentes del aerogenerador y el uso de otros algoritmos no planteados en este proyecto.

10. Bibliografía

Pineda Carlos. (2022) Aprendizaje Automático y Profundo en Python.

Canteli, M. M. (2013). D e p a r t a m e n t o de i n g e n i y r í a e l é c t r i c a y e n y r g é t i c a. Universidad de Cantabria. Fax, 942, 201385. http://www.diee.unican.es/diee/mmc_1.htmlmananam@unican.es

Capuma Condori, O. O. (2018). Convertidores AC-AC. En ELECTRONICA DE POTENCIA I.

Características Técnicas Aerogenerador Gw70 (2019). | PDF | Turbina eólica | Máquinas. Recuperado el 23 de octubre de 2022, de https://es.scribd.com/document/338444922/Caracteristicas-Tecnicas-Aerogenerador-Gw70

S, Carlos. (2017). Central Eólica Villonaco Energías Renovables, Sustentables, y Sostenibles.

CELEC EP GENSUR. (2022). Central Eólica Villonaco - Google Maps - Recuperado el 12 de diciembre de 2022, de https://www.google.com/maps/place/Central+E%C3%B3lica+Villonaco+-+CELEC+EP+GENSUR/@-4.0009927,-79.2609986,879m/data=!3m1!1e3!4m5!3m4!1s0x91cb49d80e01eb4d:0xb3c04d7a93c9697b!8m2!3d-4.0009927!4d-79.2592105!5m1!1e4

Ministerio de Energía y Minas. (2022). CENTRAL EÓLICA "VILLONACO" Recuperado el 12 de diciembre de 2022, de https://www.recursosyenergia.gob.ec/central-eolica-villonaco/

Navas Muriel, A. (2021). Análisis del uso de técnicas de Machine Learning para la identificación automática de fallas en la prescripción de antibióticos. https://bibliotecadigital.udea.edu.co/handle/10495/21864

Dhanraj, J. A., Sugumaran, V., & Joshuva, A. (2017). A Comparative Study of Bayes Classifiers for Blade Fault Diagnosis in Wind Turbines through Vibration Signals Pump condition monitoring View project Fault diagnosis of wind turbine View project A Comparative Study of Bayes Classifiers for Blade Fault Diagnosis in Wind Turbines through Vibration Signals. SDHM, 12(1), 69–90. https://www.researchgate.net/publication/321477139

C. R., & Rodrigo Jhonnatan, A. (2022). Desarrollo de un sistema de predicción de falla de cojinete de turbina hidráulica basado en redes neuronales recurrentes con LSTM.

ROMERO LOZANO, LUIS - Google Libros. (2016). Gestión del mantenimiento de instalaciones de energía eólica - Recuperado el 9 de noviembre de 2022, de https://books.google.com.ec/books?id=eR_dDQAAQBAJ&printsec=frontcover&dq=tipos+de+mantenimiento+aerogeneradores&hl=es&sa=X&redir_esc=y#v=onepage&q=tipos%20de%20mantenimiento%20aerogeneradores&f=false

Ghassemi, M. (2018). PD measurements, failure analysis, and control in high-power IGBT modules. High Voltage, 3(3), 170–178. https://doi.org/10.1049/HVE.2017.0186

Goldwind GW82/1500. (2022) - Fabricantes y aerogeneradores - Acceso en línea - The Wind Power. (2022). Recuperado el 14 de febrero de 2023, de https://www.thewindpower.net/turbine_es_441_goldwind_gw82-1500.php

Han, J., Kamber, M., & Pei, J. (2011). Data Transformation by Normalization. Data Mining: Concepts and Techniques, 113–115. https://doi.org/10.1016/B978-0-12-381479-1.00001-0

Villarubia López (2012). Ingeniería de la energía eólica - Recuperado el 4 de noviembre de 2022, de https://books.google.es/books?hl=es&lr=&id=GW_jEgJJSdcC&oi=fnd&pg=PA4&dq=aerogeneradores+e%C3%B3licos&ots=QbDTe2rqtD&sig=O7qJy3_KuyeJLKH6S16yKAoRzaA#v=onepage&q=aerogeneradores%20e%C3%B3licos&f=false

Labañino Urbina, S., Alberto, H., Zayas, V., Grabiel, O., & López, T. (2019). Random Forests Algorithm for fail detecting on computer´s networks. 12(8). http://seriecientifica.uci.cu

Liton Hossain, M., Abu-Siada, A., & Muyeen, S. M. (2018). Methods for advanced wind turbine condition monitoring and early diagnosis: A literature review. Energies, 11(5). https://doi.org/10.3390/EN11051309

Lorenzi, S. (2002). Módulos IGBT. Mundo electrónico, ISSN 0300-3787, No 333, 2002, págs. 42-47, 333, 42–47. https://dialnet.unirioja.es/servlet/articulo?codigo=255693&info=resumen&idioma=SPA

Machine Learning (2019). Recuperado el 22 de octubre de 2022, de https://iaarbook.github.io/ML/

Maldonado, J., Álvarez, O., Montaño, T., & Tenechagua, L. (2015). Análisis Climático de la Velocidad del Viento en la Región Sur del Ecuador. Revista Politécnica,

35(3), 137–137. https://revistapolitecnica.epn.edu.ec/ojs2/index.php/revista_politecnica2/article/view/402

Narvaez, J. (2021). ENERGIA EOLICA. Random Forest, el poder del Ensamble | Aprende Machine Learning. (2021). Recuperado el 1 de febrero de 2023, de https://www.aprendemachinelearning.com/random-forest-el-poder-del-ensamble/

Aguiar Dias (1998) RESISTORES.

Riquelme, E. (2006). Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial Asociación Española para la Inteligencia Artificial. Revista Iberoamericana de Inteligencia Artificial, 10, 11–18. http://www.redalyc.org/articulo.oa?id=92502902

Rubio Peña, L. (2012). Diodos.

Russell, R. (2018). Machine Learning Guía Paso a Paso Para Implementar Algoritmos De Machine Learning Con Python.

Aquilino Rodríguez Penin (2012) Sistemas SCADA . Recuperado el 22 de octubre de 2022, de https://books.google.es/books?hl=es&lr=&id=cNQfjbBcUq8C&oi=fnd&pg=PA1&dq=sistemas+scada&ots=4HQTsCMUXx&sig=q9jhmm3ByKPgnskNn7TXur4lGbY#v=onepage&q=sistemas%20scada&f=false

Suarez, K., & Carbajal, R. (2006). UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO ENERGÍA EÓLICA ENERGÍA EÓLICA ENERGÍA EÓLICA.

Tecnología del mantenimiento industrial - Félix Cesáreo Gómez de León - Google Libros. (1998). Recuperado el 22 de octubre de 2022, de https://books.google.es/books?hl=es&lr=&id=bOrFC3532MEC&oi=fnd&pg=PA21&dq=tipos+de+mantenimiento+industrial&ots=6Of-JJRnJM&sig=IlwfwcwPUmJOiyJfzSMtSCBuZmU#v=onepage&q&f=false

Valencia Ordoñez, U. P. (2021). TFM-2086 VALENCIA ORDOÃ_EZ, ULICES PAUL [Escuela Técnica Superior de Sevilla].

Velandia-Cardenas, C., Vidal, Y., & Pozo, F. (2021). Wind turbine fault detection using highly imbalanced real scada data. Energies, 14(6). https://doi.org/10.3390/en14061728

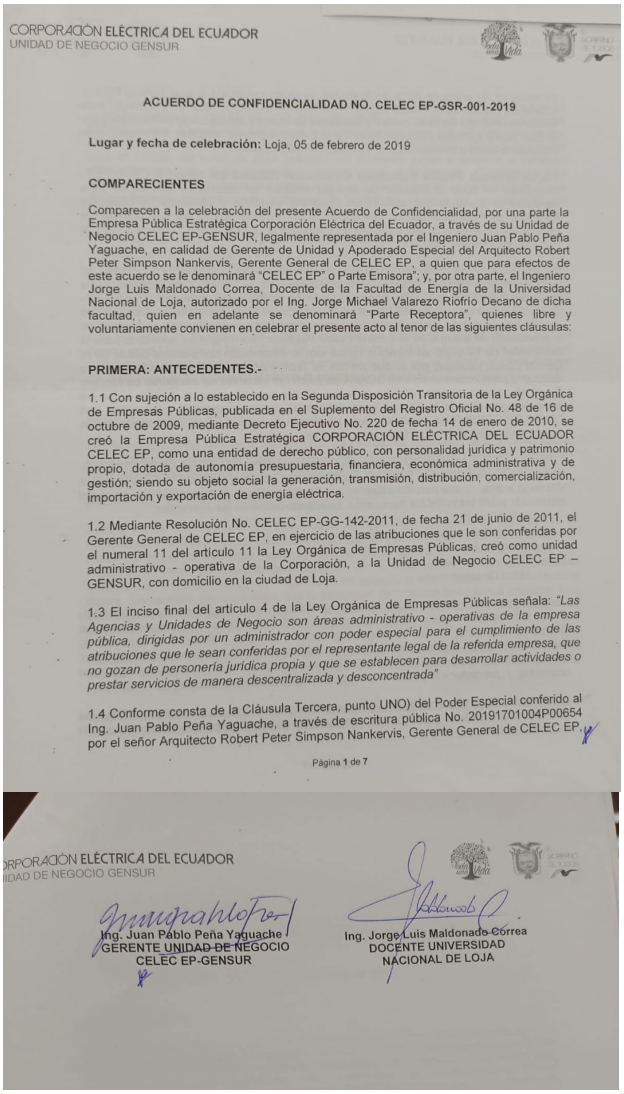
Xie, Y., Zhao, J., Qiang, B., Mi, L., Tang, C., & Li, L. (2021). Attention mechanism-based CNN-LSTM model for wind turbine fault prediction using SSN ontology annotation. Wireless Communications and Mobile Computing, 2021.

https://doi.org/10.1155/2021/6627588

Zunzunegui Fernández, B. (2021). Diseño y desarrollo de double-pulse test para módulos de IGBT.

11. Anexos

Anexo 1 Carta de confidencialidad



CORPORACIÓN ELÉCTRICA DEL ECUADOR
UNIDAD DE NEGOCIO GENSUR

ACUERDO DE CONFIDENCIALIDAD NO. CELEC EP-GSR-001-2019

Lugar y fecha de celebración: Loja, 05 de febrero de 2019

COMPARECIENTES

Comparecen a la celebración del presente Acuerdo de Confidencialidad, por una parte la Empresa Pública Estratégica Corporación Eléctrica del Ecuador, a través de su Unidad de Negocio CELEC EP-GENSUR, legalmente representada por el Ingeniero Juan Pablo Peña Yaguache, en calidad de Gerente de Unidad y Apoderado Especial del Arquitecto Robert Peter Simpson Nankervis, Gerente General de CELEC EP, a quien que para efectos de este acuerdo se le denominará "CELEC EP" o Parte Emisora"; y, por otra parte, el Ingeniero Jorge Luis Maldonado Correa, Docente de la Facultad de Energía de la Universidad Nacional de Loja, autorizado por el Ing. Jorge Michael Valarezo Riofrío Decano de dicha facultad, quien en adelante se denominará "Parte Receptora", quienes libre y voluntariamente convienen en celebrar el presente acto al tenor de las siguientes cláusulas:

PRIMERA: ANTECEDENTES.-

1.1 Con sujeción a lo establecido en la Segunda Disposición Transitoria de la Ley Orgánica de Empresas Públicas, publicada en el Suplemento del Registro Oficial No. 48 de 16 de octubre de 2009, mediante Decreto Ejecutivo No. 220 de fecha 14 de enero de 2010, se creó la Empresa Pública Estratégica CORPORACIÓN ELÉCTRICA DEL ECUADOR CELEC EP, como una entidad de derecho público, con personalidad jurídica y patrimonio propio, dotada de autonomía presupuestaria, financiera, económica administrativa y de gestión; siendo su objeto social la generación, transmisión, distribución, comercialización, importación y exportación de energía eléctrica.

1.2 Mediante Resolución No. CELEC EP-GG-142-2011, de fecha 21 de junio de 2011, el Gerente General de CELEC EP, en ejercicio de las atribuciones que le son conferidas por el numeral 11 del artículo 11 la Ley Orgánica de Empresas Públicas, creó como unidad administrativo - operativa de la Corporación, a la Unidad de Negocio CELEC EP – GENSUR, con domicilio en la ciudad de Loja.

1.3 El inciso final del artículo 4 de la Ley Orgánica de Empresas Públicas señala: *"Las Agencias y Unidades de Negocio son áreas administrativo - operativas de la empresa pública, dirigidas por un administrador con poder especial para el cumplimiento de las atribuciones que le sean conferidas por el representante legal de la referida empresa, que no gozan de personería jurídica propia y que se establecen para desarrollar actividades o prestar servicios de manera descentralizada y desconcentrada"*

1.4 Conforme consta de la Cláusula Tercera, punto UNO) del Poder Especial conferido al Ing. Juan Pablo Peña Yaguache, a través de escritura pública No. 20191701004P00654 por el señor Arquitecto Robert Peter Simpson Nankervis, Gerente General de CELEC EP,

Página 1 de 7

ORPORACIÓN ELÉCTRICA DEL ECUADOR
IDAD DE NEGOCIO GENSUR

Ing. Juan Pablo Peña Yaguache
GERENTE UNIDAD DE NEGOCIO
CELEC EP-GENSUR

Ing. Jorge Luis Maldonado Correa
DOCENTE UNIVERSIDAD
NACIONAL DE LOJA

Anexo 2 Certificado de traducción del resumen.

Loja, 26 de abril 2024

Magister
JHIMI BOLTER VIVANCO LOAIZA
CATEDRÁTICO DE LA CARRERA DE PEDAGOGÍA DE LOS IDIOMAS NACIONALES Y EXTRANJEROS - UNL

C E R T I F I C O:

Que el documento aquí expuesto es fiel traducción del idioma español al idioma inglés del resumen del Trabajo de Integración Curricular titulado: **"Uso de técnicas de Machine Learning para la detección de fallas en el módulo IGBT de los aerogeneradores del Parque Eólico Villonaco".**, de autoría de Pablo José Caraguay Quinde, con cédula de identidad 1105656134, de la Carrera de Ingeniería Electromecánica de la Universidad Nacional de Loja.

Lo certifico y autorizo hacer uso del presente en lo que a sus intereses convenga.

Firmado electrónicamente por: JHIMI BOLTER VIVANCO LOAIZA

JHIMI BOLTER VIVANCO LOAIZA, M.Ed.
CATEDRÁTICO DE LA CARRERA DE PEDAGOGÍA DE LOS IDIOMAS NACIONALES Y EXTRANJEROS - UNL