UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE CUENCA

CARRERA DE INGENIERÍA MECATRÓNICA

DIAGNÓSTICO DEL NIVEL DE SEVERIDAD DE FALLO DE DIENTE ROTO EN ENGRANAJES RECTOS USANDO ALGORITMOS DE APRENDIZAJE AUTOMÁTICO Y GRÁFICAS DE POINCARÉ APLICADOS A LA SEÑAL DE PAR ELÉCTRICO DE UN MOTOR DE INDUCCIÓN

Trabajo de titulación previo a la obtención del título de Ingeniera Mecatrónica

AUTOR: MARÍA VICTORIA MEJÍA CORREA

TUTOR: ING. MARIELA CERRADA, Ph.D.

Cuenca – Ecuador

2022

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Yo, María Victoria Mejía Correa con documento de identificación N° 0107577892 manifiesto que:

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

Cuenca, 29 de julio del 2022

Atentamente,

María Victoria Mejía Correa
0107577892

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

Yo, María Victoria Mejía con documento de identificación N° 0107577892, expreso mi voluntad y por medio del presente documento cedo a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que soy autor del Proyecto Técnico: "Diagnóstico del nivel de severidad de fallo de diente roto en engranajes rectos usando algoritmos de aprendizaje automático y gráficas de Poincaré aplicados a la señal de par eléctrico de un motor de inducción", el cual ha sido desarrollado para optar por el título de: Ingeniera Mecatrónica, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

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

Cuenca, 29 de julio del 2022

Atentamente,

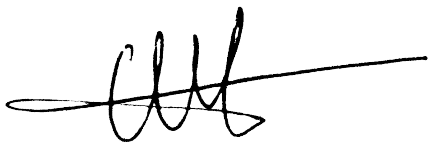
María Victoria Mejía Correa
0107577892

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN

Yo, Mariela Cerrada Lozada con documento de identificación N° 0151771813, docente de la Universidad Politécnica Salesiana, declaro que bajo mi tutoría fue desarrollado el trabajo de titulación: DIAGNÓSTICO DEL NIVEL DE SEVERIDAD DE FALLO DE DIENTE ROTO EN ENGRANAJES RECTOS USANDO ALGORITMOS DE APRENDIZAJE AUTOMÁTICO Y GRÁFICAS DE POINCARÉ APLICADOS A LA SEÑAL DE PAR ELÉCTRICO DE UN MOTOR DE INDUCCIÓN, realizado por María Victoria Mejía Correa con documento de identificación N° 0107577892, obteniendo como resultado final el trabajo de titulación bajo la opción Proyecto Técnico que cumple con todos los requisitos determinados por la Universidad Politécnica Salesiana.

Cuenca, 29 de julio del 2022

Atentamente,



Ing. Mariela Cerrada Lozada, Ph.D
0151771813

Dedicatoria

El presente proyecto de titulación está dedicado:

A mi familia, a todas las personas que formaron parte del proceso, a los estudiantes e investigadores que puedan hacer uso del conocimiento presentado en este trabajo de titulación.

Agradecimientos

Agradezco a mis padres por permitirme crecer como persona y profesional. Su apoyo y confianza ha sido fundamentales para concluir la carrera y cumplir una meta más. El desarrollo de este trabajo no hubiera sido posible sin su motivación y palabras de aliento cuando más las necesitaba.

También agradezco a mi Tutora de Proyecto de Titulación Mariela Cerrada Ph. D. por saber orientarme a largo de este proceso y darme la oportunidad de desarrollarme en áreas como la investigación dentro del grupo GIDTEC.

Este documento fue realizado enteramente en LaTeX

Índice

Lista de Tablas

Lista de Figuras

Resumen

En este documento titulado “Diagnóstico del nivel de severidad de fallo de diente roto en engranajes rectos usando algoritmos de aprendizaje automático y gráficas de Poincaré aplicados a la señal de par eléctrico de un motor de inducción” tiene como enfoque principal determinar el mejor algoritmo de aprendizaje automático entre K vecinos más cércanos (*K-Nearest Neightbor* en inglés) y Bosques aleatorios (*Random Forest* en inglés) y también las mejores características, o atributos, de Poincaré usadas para el entrenamiento de los modelos de clasificación. El objetivo de la implementación de estos algoritmos es obtener modelos computacionales con métricas de valores deseados, por ejemplo, valores de precisión y exactitud altos que permitan determinar el nivel de severidad de diente roto a partir de la señal de par eléctrico del motor que actúa sobre la caja de engranajes que pertenece a un prototipo disponible en el Laboratorio de Vibraciones de la Universidad Politécnica Salesiana, Sede Cuenca, Ecuador.

La sección 1 del documento presenta la introducción, en la cual se describe los aspectos generales del Manteniendo Basado en la Condición (MBC) y cómo este puede ser implementado mediante los algoritmos mencionados al usar la señal del par eléctrico. La sección 2 indica el problema y los antecedentes relacionados con el trabajo, así como, su importancia, sus alcances y su delimitación. Los objetivos planteados para este proyecto se presentan en la sección 3 y son descritos tanto de forma general, como específica. Por otro lado, el marco teórico del trabajo se presenta en la sección 4, en donde se abarcan sobre temas generales del trabajo como: el MBC, la obtención del par eléctrico, los fallos que se producen en una caja de engranajes, la banco de pruebas del laboratorio de la Universidad Politécnica Salesiana, entre otros. En cuanto al enfoque principal del trabajo, se detallan los aspectos teóricos sobre los diagramas de Poincaré (DP), y la extracción de características o atributos. Por último, se explica sobre los algoritmos *K-Nearest Neightbor* (KNN) y *Random Forest* (RF) y sus respectivos parámetros a considerar.

Por otro lado, en la sección 5, el marco metodológico es descrito detalladamente. Se explica el desarrollo de modelos capaces de clasificar los diferentes niveles de severidad de diente roto al usar algoritmos como KNN y RF. Después se describe el procesamiento de la base de datos del par eléctrico y la extracción y selección de características a partir de diagramas de Poincaré. Las características o atributos mencionados fueron obtenidos de un análisis exploratorio de datos, el cual fue un trabajo de investigación realizada previamente por la autora. En relación

con los algoritmos implementados, para el caso de KNN se usaron distancias como: Coseno, Euclidiana y Mahalanobis. Además, se tomó la cuenta la validación cruzada de este modelo y el número de vecinos. Para RF, se consideraron aspectos como el número de árboles, número de iteraciones, entre otros. Los pasos seguidos para determinar el mejor modelo y los mejores atributos se explica dentro de esta metodología.

En la sección 6 se describen los resultados obtenidos después de haber seguido la metodología explicada. Se calcularon las métricas de precisión, exactitud, recall y F1-score, los cuales son indicadores del rendimiento de cada uno de los modelos computacionales generados. Los resultados están enfocados a las métricas mencionadas y para resumirlos se realizaron gráficas 3D, diagramas de barras y tablas donde se presentan los mejores resultados según algoritmo implementado en los modelos y sus respectivos parámetros seleccionados. Finalmente, en la sección 7, se muestran las conclusiones y recomendaciones finales de este trabajo de titulación. Aquí se menciona las condiciones para obtener la mejor clasificación a partir de los resultados obtenidos y al mismo tiempo, se indica otro tipo de trabajos a realizar en el futuro para expandir la aplicación del par eléctrico como señal para el MBC.

Palabras clave: Aprendizaje Automático, Atributos de Poincaré, K-Nearest Neighbors, Random Forest, Caja de engranajes, Diente roto, Nivel de severidad de fallo, Par eléctrico.

Abstract

In this document entitled "Diagnosis of the severity level of broken tooth failure in spur gears using machine learning algorithms and Poincaré graphs applied to the electrical torque signal of an induction motorïts main focus is to determine the best Machine Learning algorithm between K nearest neighbors and Random Forests and also the best Poincaré features, or attributes, used for training the classification models. The objective of the implementation of these algorithms is to obtain computational models with metrics of desired values, for example, high precision and accuracy values that allow determining the level of severity of broken tooth from the electric torque signal of the motor that acts on a gearbox that belongs to a prototype available at the Vibration Laboratory of the Salesian Polytechnic University, Cuenca, Ecuador.

Section 1 of the document presents the introduction, in which the general aspects of Condition Based Maintenance (CBM) are described and how it can be implemented through the mentioned algorithms when using the electrical torque signal. Section 2 indicates the problem and the background related to the work, as well as its importance, its scope and its delimitation. The objectives set for this project are presented in section 3 and are described both in a general and specific way. On the other hand, the theoretical framework of the work is presented in section 4, where general topics of the work are covered, such as: the MBC, obtaining the electrical torque, the failures that occur in a gearbox, the bench of laboratory tests of the Salesian Polytechnic University, among others. Regarding the main focus of the work, the theoretical aspects of Poincaré diagrams (DP) and the extraction of characteristics or attributes are detailed. Finally, it is explained about the *K-Nearest Neightbor* (KNN) and *Random Forest* (RF) algorithms and their respective parameters to consider.

On the other hand, in section 5, the methodological framework is described in detail. The development of models capable of classifying the different levels of severity of broken teeth using algorithms such as KNN and RF is explained. Then, the processing of the electrical torque database and the extraction and selection of features from Poincaré diagrams are described. The characteristics or attributes mentioned were obtained from an exploratory data analysis, which was a research work previously carried out by the author. In relation to the implemented algorithms, in the case of KNN, distances such as: Cosine, Euclidean and Mahalanobis were used. In addition, the cross-validation of this model and the number of neighbors were taken into account. For RF, aspects such as the number of trees, number of iterations, among others, were considered. The steps followed to determine the best model

and the best attributes are explained within this methodology.

Section 6 describes the results obtained after having followed the explained methodology. The precision, accuracy, recall and F1-score metrics were calculated, which are indicators of the performance of each of the generated computational models. The results are focused on the metrics mentioned and to summarize them, 3D graphs, bar diagrams and tables are made where the best results are presented according to the algorithm implemented in the models and their respective selected parameters. Finally, in section 7, the final conclusions and recommendations of this degree work are shown. Here the conditions to obtain the best classification from the results obtained are mentioned and at the same time, other types of work to be carried out in the future to expand the application of the electrical torque as a signal for the MBC are indicated.

Keywords: Machine Learning, Poincaré Attributes, K-Nearest Neighbors, Random Forest, Gearbox, Broken Tooth, Failure Severity Level, Electrical Torque.

1. Introducción

El mantenimiento adecuado de equipos industriales permite reducir costos, puesto que dentro de la producción de empresas o fabricas este puede representar entre 15 % y 70 % de los costos totales, considerándolo como uno de los más relevantes (Al-Najjar, Ingwald, y Kans, 2016). Existen diferentes tipos de mantenimiento como el Mantenimiento Correctivo, Mantenimiento Predictivo y Mantenimiento Preventivo. Para el caso de este proyecto se hará uso de una de las ramas del manteamiento Predictivo, que es el Mantenimiento Basado en la Condición (MBC), el cual indica si existen comportamientos no deseados en el funcionamiento de la maquinaria o equipo monitoreado a partir de un análisis de datos (Bevilacqua y Braglia, 2000; Coria, Maximov, Rivas-Dávalos, Melchor, y Guardado, 2015). Mucha de la maquinaria industrial posee cajas de engranajes debido a sus cualidades para transmitir movimiento y potencia entre ejes. El mantenimiento de caja de engranajes es necesario para evitar el paro de maquinaria y perdida en la producción. De este modo, la implementación de un programa de mantenimiento de este tipo es posible dentro de este contexto. El MBC consiste en monitorear el sistema para diagnosticar y detectar fallos de un componente en específico antes que este se convierta en una falla funcional. Para aplicar esta técnica de mantenimiento, se debe realizar lo siguiente (Ahmad y Kamaruddin, 2012; Jardine, Lin, y Banjevic, 2006):

1. Adquisición de datos: Medición, obtención y almacenamiento de información de relevante. Ejemplos de esto pueden incluir señales de corriente, acústicas, entre otras.

2. Procesamiento de datos: Permite realizar un análisis de la información adquirida y la obtención características que permiten una interpretación del comportamiento de los datos. Las características en general se relacionan con valores estadísticos calculados sobre la señal monitoreada como el promedio, desviación estándar y muchos más (Sánchez Loja, 2018).

3. Toma de decisiones: A partir del diagnóstico y/o pronóstico de la maquinaria se determina las acciones que se llevaran a cabo para solucionar los fallos y dar la atención requerida a la maquinaria.

Una forma del aplicar el MBC es el uso del aprendizaje automático, el cual es una rama de la inteligencia artificial. Debido a que se procesan datos y se los selecciona de manera adecuada, se puede hacer el uso de modelos computacionales generados al implementar algoritmos apropiados, permiten clasificar las posibles fallas y conocer de ese modo el estado del equipo tratado (R. Liu, Yang, Zio, y Chen, 2018). A lo largo de este trabajo de titulación

se utilizó la señal del par eléctrico de un motor que actúa sobre una caja de engranajes porque es sensible al fallo de diente roto y a sus diferentes niveles de severidad (Ortega Lucero, 2021). En un trabajo realizado con anterioridad por la autora, se extrajeron indicadores estadísticos de la señal del par eléctrico que permiten el entrenamiento de un modelo de aprendizaje automático (Mejía, 2022).

Los indicadores usados fueron los atributos de Poincaré ordenados según su importancia de acuerdo con diferentes métodos como: Composing Density Between and With clusters (CDbw) y ReliefF. Además, en este trabajo, se extendió el uso del modelo generado por RF para proveer otro método adicional que permite ordenar según su importancia los atributos de Poincaré extraídos (en inglés este método es conocido como *Ranking*). Los diferentes conjuntos de atributos se usaron como entradas para los algoritmos KNN y RF con el fin de estimar el nivel de rotura de diente de engranes mediante la generación de modelos computacionales.

Los algoritmos KNN y RF son algoritmos robustos al momento de clasificar muestras, es decir, tienen un alto rendimiento y debido a su funcionamiento (KNN basado en la distancia entre muestras y RF en árboles de decisiones) son adecuados para el desarrollo de este proyecto. Mediante estos algoritmos se generaron diferentes modelos con determinados parámetros. Una vez con los modelos entrenados, se ingresan nuevos datos de la señal del par eléctrico para conocer sus respectivas capacidades de clasificación a partir de métricas como la exactitud, precisión, recall y F1-score en la clasificación de los datos de prueba. De esta manera es posible determinar el nivel de diente roto. Los modelos obtenidos fueron usados para clasificar datos de entrenamiento y de prueba y al realizar un análisis de estos resultados, se identificaron cuáles fueron los atributos y parámetros que influyen en el rendimiento. De este modo, se utilizó dos algoritmos que permiten generar modelos computacionales con la capacidad de diagnosticar el grado de severidad de diente roto de una caja de engranajes rectos.

2. Problema

2.1. Antecedentes

Toda maquinaria requiere de su determinado mantenimiento con el fin prolongar su vida útil y minimizar pérdidas económicas dentro de una empresa. Existen diferentes tipos de mantenimiento, y para llevar a cabo un mantenimiento adecuado es fundamental conocer el estado de salud de la maquinaria y de esa manera, conservar la eficiencia y efectividad dentro

de una planta. Una vez comprendido este contexto, varias han sido las empresas que han optado por desarrollar procesos y métodos para detección de fallos.

En sus principios, las estrategias de mantenimiento se basaban en el mantenimiento correctivo, el cual era realizado cuando los fallos ya se presentaban y esto significaba inconvenientes dentro de la producción, por ejemplo contratiempos al mantener la maquinaria parada y diversos riesgos o accidentes que pueden producirse al no trabajar con la maquinaria en condiciones óptimas. Hoy en día, el mantenimiento preventivo es el más usado, generalmente, dentro de la industria porque consiste en remplazar los elementos o componentes en relación con tiempo definido o la experiencia requerida al usar dicho componente y cambiarlo; sin embargo, estos pueden ser cambiando cuando todavía se encuentran en buen estado. Por otro lado, el mantenimiento predictivo realiza las acciones de mantenimiento hasta que sea verdaderamente requerido, porque se lleva un constante monitoreo del sistema para identificar fallos o condiciones anormales cuando está funcionando el equipo. Estas condiciones se las suele asociar a daños en el elemento; por lo general se monitorean señales de vibración, corriente, entre otras. El mantenimiento predictivo llega a abarcar el Mantenimiento Centrado en la Confiabilidad (MCC) y también el Mantenimiento Basado en la Condición (MBC) (Sánchez Loja, 2018); este último mencionado es el que enmarca la propuesta de este trabajo.

A lo largo de este documento, el enfoque es hacia la maquinaria rotativa, específicamente a las cajas de engranajes. Se debe mencionar la importancia de las mismas, por la transición del par mecánico y su capacidad de aumentar o disminuir la potencia. Al ser los engranajes elementos importantes del funcionamiento de la maquinaria, es necesario conocer el estado de los mismos y de ese modo, es posible implementar un sistema capaz de detectar diferentes niveles de diente roto y así generar un programa de MBC con el fin de que los gastos en producción sean menores y la calidad de los productos o servicios sea alta (Al-Najjar y cols., 2016). Como se ha mencionado, para detectar diferentes niveles de rotura de diente en una caja de engranajes es posible usar señales tales como: vibración, corriente, emisión acústica, entre otros.

Para el desarrollo de este trabajo se utilizó la señal del par eléctrico porque en otros trabajos, como el de Ortega Lucero (2021) se ha demostrado que la señal de par eléctrico (fuerza generada por el motor en su eje) de una máquina eléctrica (específicamente un motor de tres fases) conectado a una caja de engranajes es una señal perceptible al fallo de diente roto. De este modo, la señal permite la detección del nivel de severidad de rotura de diente en una caja de engranajes rectos. Una de las principales razones para usar la señal del par

eléctrico es que permite detectar fallos en una etapa inicial y, si se la compara con señales de vibración, no es necesario colocar sensores dentro de la caja de engranes, debido a que solo se requiere conocer las señales de la corriente y el voltaje del motor que actúa sobre la maquinaria (Ortega Lucero, 2021). Al medir estas señales, se puede determinar el par eléctrico, el cual es usado para obtener sus atributos o características de Poincaré y al usarlas en modelos de aprendizaje automático para identificar patrones de fallos.

Adicionalmente, en un trabajo previo de pasantías de investigación de la autora, se determinó a través de un análisis exploratorio de datos que las características de Poincaré extraídas del par eléctrico pueden ser discriminativas en el problema de clasificación de nivel de severidad de diente roto. Dentro de dicho trabajo se realizó el *Ranking* de atributos de Poincaré con ReliefF y CDbw para comparar sus resultados mediante gráficas t-SNE, diagramas de caja (*boxplotting* en inglés), entre otros (Mejía, 2022). Igualmente, en el trabajo de Loaiza Sánchez (2021) se implementaron las características de Poincaré aplicadas a señales de vibración que fueron usadas para generar modelos de clasificación con el algoritmo LAMDA HAD. Mediante estos dos trabajos mencionados, se concluyo que los atributos de Poincaré son una herramienta útil para calcular indicadores estadísticos discriminativos de diferentes modos de fallo al usar señales de vibración y que las características de Poincaré de la señal del par eléctrico discriminan el nivel de severidad diente roto. De esta forma, se considera que es posible realizar la propuesta del presente trabajo de titulación para generar modelos de clasificación usando KNN y RF.

2.2. Descripción del problema

Las cajas de engranes en muchos de los casos están cerradas, lo cual dificulta inspeccionarlas directamente y detectar los posibles daños generados. Algunos de los fallos que se suelen generar en cajas de engranes incluyen diente roto, grietas y picadura en el elemento y por último el gripado. De este modo, para evitar el paro de la maquinaria, se opta por técnicas de monitoreo de la condición. Se debe mencionar que mediante esta técnica se puede obtener diferentes señales que indican el estado de salud de la maquinaria (Loaiza Sánchez, 2021). Llegar a clasificar la severidad de fallos en cajas de engranajes rectos con base al análisis de determinadas señales (específicamente par eléctrico y la rotura de diente) es de enfoque principal de este proyecto.

Los modos de fallos que llegan a ocurrir en una caja de engranajes dan complicaciones

al momento de detectarlos y solamente se nota cuando ya se ha producido la falla o avería de la caja de engranajes. Es debido a esta razón, que al aplicar técnicas de monitoreo de condición es posible la obtención de diferentes señales que muestre cuál es el estado de salud de la maquinaria (Sánchez Loja, 2018). Las señales que pueden analizarse al realizar un mantenimiento, como el mencionado a lo largo del documento, abarcan señales de audio, vibración y otras; sin embargo, también se encuentra la señal de par eléctrico, la cual no es usada en la detección de fallos generalmente. De todas maneras, esta señal ha demostrado tener sensibilidad a diferentes grados de severidad de diente roto (Ortega Lucero, 2021), por lo tanto, representa una nueva alternativa para diagnosticar el estado de maquinaria rotativa.

Las señales de par eléctrico fueron obtenidas mediante un modelo matemático que usa señales tanto de corriente y de voltaje. Estas señales fueron medidas y adquiridas por el grupo GIDTEC en un banco de pruebas del laboratorio de vibraciones de la Universidad Politécnica Salesiana. A partir de este modelo y del trabajo realizado por Ortega Lucero (2021), se obtuvieron las señales del par eléctrico para cada nivel de severidad de diente roto y una vez con estas señales definidas, se realizó el pertinente procesamiento de los datos para que sean ingresados a modelos computacionales que clasifican diferentes grados de severidad de rotura de diente roto en una caja de engranes. Para lograr lo mencionado, se aplica algoritmos de aprendizaje automático y los atributos de Poincaré calculados previamente (Medina y cols., 2017).

2.3. Importancia y alcances

El MBC es una estrategia de mantenimiento que basa sus resultados en la disponibilidad, fiabilidad, coste de mantenimiento, vida útil de la instalación, seguridad y bajo impacto ambiental, en el diagnóstico previo de los equipos (R, S, y W, 2006). Para el caso de este proyecto, el enfoque es hacia la maquinaria rotativa que tiene cajas de engranajes. Se ha mencionado que existen diferentes señales monitoreadas que se pueden medir de una caja de engranajes, y específicamente tomando las señales del par eléctrico del motor que actúa sobre ella, es posible proporcionar una nueva forma para detectar fallos en cajas de engranajes y en este aspecto, se aplica específicamente para el fallo de diente roto (Ortega Lucero, 2021). Para este proyecto, el par eléctrico se obtuvo mediante la corriente y el voltaje medidos del motor trifásico acoplado a la caja de engranes y las mediciones se realizaron mediante sensores de fácil emplazamiento en la maquinaria, lo cual a diferencia del monitoreo de vibraciones no es

invasivo, puesto que en los otros casos es necesario disponer de sensores dentro de la caja de engranes (Loaiza Sánchez, 2021).

Actualmente, existen diferentes tipos de mantenimiento, por lo tanto, es recomendable el establecimiento de un programa de mantenimiento integral que incluya técnicas de monitoreo y diagnóstico mediante pruebas no destructivas, ni invasivas, con el fin que de que la disponibilidad y seguridad de la planta sea alta y los costes bajos (Sánchez Loja, 2018). Por otro lado, el MBC sirve para evitar fallas catastróficas de máquinas rotativas con un estado crítico y por esta razón, el uso del par eléctrico en estrategias MBC resultada adecuada para la detección del nivel de severidad de rotura de diente en una caja de engranes. Al usar la señal del par eléctrico en algoritmos de aprendizaje automático, existe la posibilidad de que los modelos computacionales generados clasifiquen los diferentes niveles de severidad por rotura de diente, lo cual, en otras palabras, permite conocer el estado de la caja de engranajes y así tomar una decisión a futuro.

Este proyecto no solo sirve como una fuente de información y conocimiento para alumnos de la Universidad Politécnica Salesiana y para los integrantes del Grupo de Investigación y Desarrollo en Tecnologías Industriales (GIDTEC), los cuales aprenderán nuevas tecnologías para solventar las necesidades técnicas, tecnológicas y de investigación en el área del mantenimiento industrial, sino también para la industria local que posee este tipo de maquinaria y que se beneficia de este aporte en posibles futuras actividades de colaboración entre la academia y la industria.

2.4. Delimitación

El problema de estudio se delimitará en las siguientes dimensiones:

2.4.1. Espacial o geográfica

Este trabajo se realizó en la Universidad Politécnica Salesiana, sede Cuenca, que se encuentra ubicada en la Calle Vieja 12-30 y Elia Liut. El plan experimental para la adquisición de señales de la caja de engranajes y la base de datos del par eléctricos están disponibles en el laboratorio de vibraciones de esta universidad, en donde se desarrollan las actividades de instigación del grupo GIDTEC, sede Cuenca, Ecuador.

Figura 1

Ubicación de la Universidad Politécnica Salesiana

Nota: *Obtenido de Google Maps (s.f.).*

2.4.2. Temporal

El tiempo trascurrido para realizar la investigación y desarrollo de este proyecto fue de 6 meses.

2.4.3. Sectorial o institucional

La Universidad Politécnica Salesiana sede Cuenca es de carácter privada y se localiza en el barrio El Vecino, parroquia El Vecino.

2.5. Problema General

- ¿Es posible mediante un modelo de aprendizaje automático detectar diferentes niveles de severidad de fallo de diente roto en una caja de engranes rectos usando la señal del par eléctrico?

2.6. Problemas Específicos

1.- ¿Cuál es el algoritmo de aprendizaje automático en un entorno supervisado que tiene mejor desempeño para la clasificación de niveles de severidad de falla de diente roto?

2.- ¿Es posible diagnosticar con precisión apropiada el grado de severidad en la rotura del diente en una caja de engranes a partir de la señal de par eléctrico usando un modelo computacional ajustado por un algoritmo de aprendizaje automático?

3.- ¿Cuáles son los atributos más representativos a extraer de las señales del par eléctrico usando diagrama de Poincaré para la detección de fallos y usarlos en el algoritmo de aprendizaje automático?

3. Objetivos

3.1. Objetivo General

- Diagnosticar diferentes niveles de severidad de diente roto en una caja de engranes a partir de análisis de la señal del par eléctrico usando algoritmos de aprendizaje automático supervisados.

3.2. Objetivos Específicos

- Aplicar al menos dos algoritmos de aprendizaje automático para determinar el nivel de severidad de fallo de diente roto en una caja de engranajes a partir de atributos de Poincaré.
- Desarrollar un estudio comparativo entre los diferentes modelos computacionales de aprendizaje automático obtenidos, con base a la precisión en clasificación.
- Determinar la influencia de atributos de Poincaré extraídos del par eléctrico, en el desempeño de los modelos de aprendizaje automático con base a la precisión en clasificación.

4. Hipótesis

4.1. Hipótesis General

- El par eléctrico es una señal sensible a fallos en una caja de engranes y puede usarse en modelos de detección de fallos ajustados con algoritmos de aprendizaje automático.

4.2. Hipótesis Específicas

1.- Al usar diferentes algoritmos de aprendizaje automático se obtienen diferentes modelos (KNN y RF), lo cual permite conocer cuál de ellos tendrá un mejor desempeño al detectar niveles de severidad de fallo de diente roto de un engranaje.

2.- Dependiendo tanto de los atributos usados como de los modelos computacionales, será posible realizar una comparación y determinar cuál de los mismos es el más preciso.

3.- Al usar los atributos extraídos de gráficos de Poincaré y seleccionados por técnicas, tales como: CDwb, Random Forest y RefliefF es posible establecer los atributos adecuados para la detección de fallos usando diferentes modelos computacionales.

5. Marco Teórico

5.1. Mantenimiento basado en la condición (MBC)

El MBC basa sus decisiones mediante el diagnóstico de los equipos. Una vez determinado su diagnóstico, se plantea si se debe actuar sobre la maquinaria cuando un elemento no está funcionando de la forma deseada, por lo tanto, esto requiere una acción específica: limpiar, apretar, engrasar, reacondicionar, sustituir y otros (Scarf, 2007). Se usan diferentes técnicas para realizar un diagnóstico, lo cual incluye la inspección visual superficial, la inspección detallada, las verificaciones de funcionamiento, el análisis de datos obtenidos con instrumentos montados en línea o el análisis de datos obtenidos con instrumentos portátiles que se instalan en un equipo durante las pruebas y después se trasladan a otros (Sánchez Loja, 2018).

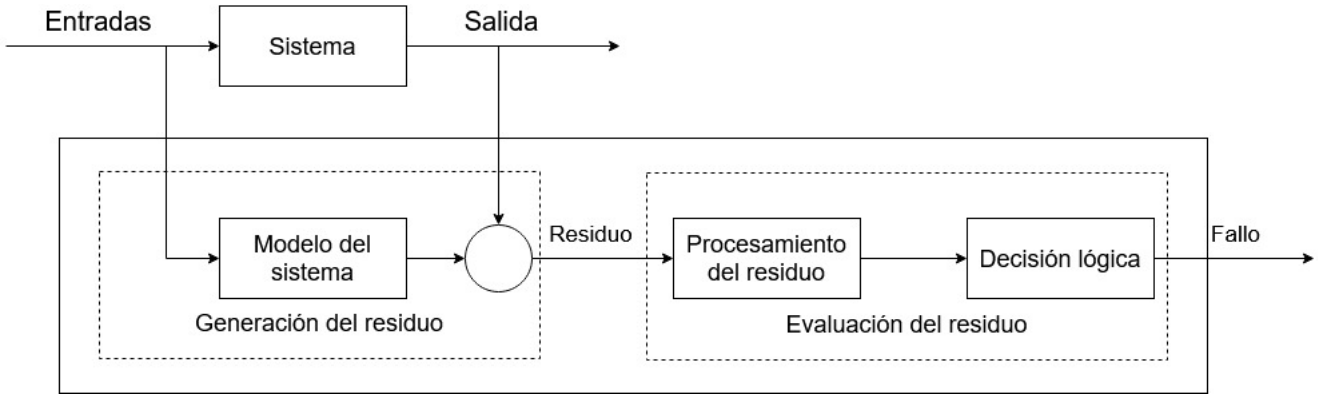
Se debe mencionar que al existir diferentes formas de realizar MBC, al monitorear el estado de la maquinaria y al usar de modelos computacionales, se puede clasificar fallos a partir del aprendizaje automático (*Machine Learning* en inglés), una vez que los modelos son entrenados y permiten determinar el rendimiento del sistema (Scarf, 2007). El diagnóstico de

fallos consta de varias tareas como la detección, el aislamiento y la identificación del fallo con en objetivo de una encontrar una relación entre los indicadores de condición con las anomalías (Bayar, Darmoul, Hajri-Gabouj, y Pierreval, 2015). Hay dos tipos de enfoques de diagnóstico de fallos que usan señales de monitoreo de condición que son: basados en modelos (*model-based* en inglés) y basados en datos (*data-based or data-driven methods* en inglés) (Kan, Tan, y Mathew, 2015).

1. **El diagnóstico basado en modelos:** Se basa en la estimación de salidas de un sistema a partir de su modelo matemático correspondiente. Además, se compara dichos valores estimados con valores reales medidos para así obtener una señal residuo que permite conocer propiedades del sistema al fallar y también permite determinar si existe un posible fallo (Isermann, 2005). La Figura 2 muestra el esquema general de este tipo de monitoreo. El conocimiento de las características físicas relacionadas con los diferentes fallos del equipo o de la maquinaria a tratar es necesario para que su respectivo modelo matemático sea capaz de describir el comportamiento del sistema y generar un diagnóstico adecuado (Scarf, 2007).

Figura 2

Esquema del diagnóstico basado en modelos .



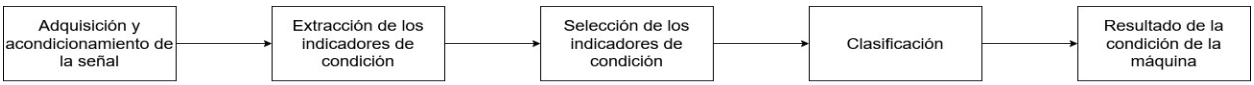
Nota: *En el esquema se observa lo diferetnes bloques perteneciciente al diagram para el diagnóstico basado en modelos (Sánchez Loja, 2018).*

2. **El diagnóstico de fallos basado en datos:** Consiste en el reconocimiento de patrones de señales monitoreadas mediante diversos métodos. Es posible extraer indicadores estadísticos de las señales mencionadas en diferentes condiciones de fallo. De este modo,

este tipo de diagnóstico posee un proceso de entrenamiento con todos los patrones según su respectiva condición de fallo, lo cual permite conocer el estado de los equipos o maquinaria (Scarf, 2007). Este tipo de diagnóstico es eficiente en cajas de engranajes y sirve como una alternativa de MBC (Loaiza Sánchez, 2021; Medina y cols., 2017). Los pasos de este proceso son mostrados en la Figura 3.

Figura 3

Esquema del diagnóstico de fallos basado en datos.



Nota: *Bloques del diagnóstico basado en datos Sánchez Loja (2018).*

5.2. Máquina de inducción

En esta sección se explica el modelo de las máquinas de inducción y su funcionamiento porque al tratarse de maquinaria rotativa dentro de la industria, los motores de tres fases son los más usados. La alimentación del motor consta de tensiones trifásicas balanceadas encontradas dentro del estator y, por otro lado, hay corriente que circula en los conductores del rotor debido a que en los devanados se produjo un campo magnético que gira constantemente, induciendo así una fuerza electromotriz (Amador, Bueno, y Amador, 2009). También se genera otro campo magnético, el cual interactúa con el estator, produciendo un par eléctrico porque hay corriente circulando por el rotor. Esta señal es la que se aplica a lo largo de este trabajo de titulación.

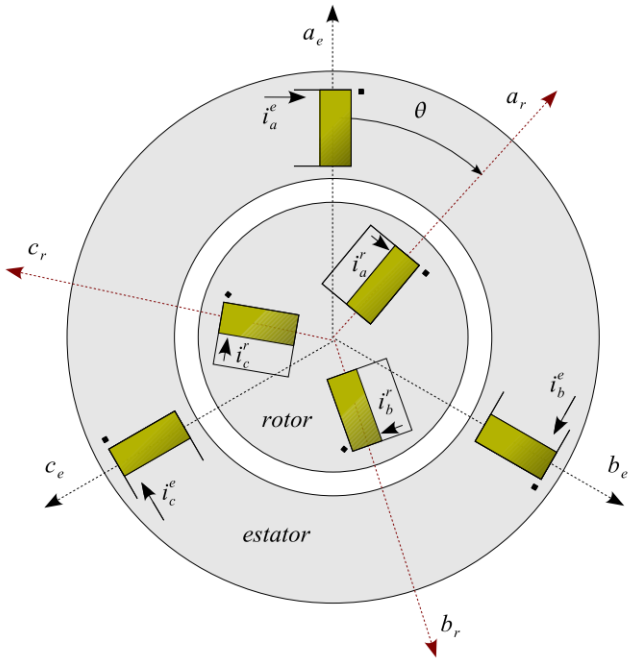
5.2.1. Par eléctrico

Es importante resaltar que la obtención del par eléctrico se justifica a partir del modelo matemático del motor trifásico. Por lo general, las máquinas de inducción tienen tres bovinas tanto en estator y rotor y a posición angular del rotor de una máquina de inducción permite definir y desarrollar un modelo matemático para el mismo en coordenadas primitivas sin existir una dependencia lineal. De esta manera es posible obtener un modelo matemático basado en ecuaciones diferenciales que describen el comportamiento dinámico de la máquina de inducción en relación con la posición angular del rotor. Se debe mencionar que existen diferentes transformaciones de coordenadas que hace más fácil el manejo de las ecuaciones y sus dimensiones (Aller, 2006). En la Figura 4 se puede observar y entender de forma gráfica

de donde se obtiene el modelo dinámico. Los ejes coordenados asociados a las fases a, b, c del motor en el estator y rotor son $a_{e,r}$, $b_{e,r}$ y $c_{e,r}$, por otro lado, la corriente de fase tanto en rotor como en el estator son $i^e_{a,b,c}$ y $i^r_{a,b,c}$. Por último, θ es la posición angular del rotor (Aller, 2006).

Figura 4

Esquema de motor trifásico.



Nota: *Máquina de induccion con sus respectivas bovinas en estator y rotor para la generación del modelo dinámico (Aller, 2006).*

Se debe mencionar que las partes extremas de los devanados del rotor están en cortocircuito y a partir de las leyes de Kirchoff es posible plantear las ecuaciones eléctricas del motor. De este modo se presenta la Ecuación (1) (Aller, 2006):

$$[v] = [R][i] + [L(\theta)]\frac{d[i]}{dt} + \dot{\theta}[\tau(\theta)][i] \tag{1}$$

donde [R] representa la matriz de resistencias eléctricas constantes del estator y rotor, en tanto que [L(θ)] es la matriz de inductancias del motor y [$\tau(\theta)$] es la primera derivada de la matriz [L(θ)]. Estas matrices dependen de la posición del eje del motor θ. La definición de

tales matrices se encuentran detalladas en Aller (2006), por lo que se sugiere revisarla para una mejor comprensión del tema. En relación con las expresiones de voltaje, corriente y el flujo del motor, se presentan las Ecuaciones (2), (3) y (4) respectivamente, a continuación:

$$[v] = \begin{bmatrix} [v_e] \\ [v_r] \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} v_a^e & v_b^e & v_c^e \end{bmatrix}^t \\ \begin{bmatrix} v_a^r & v_b^r & v_c^r \end{bmatrix}^t \end{bmatrix} \tag{2}$$

donde la variable v representa el vector de voltaje, v_e representa el voltaje en el estator de forma de vector, así como v_r es el voltaje en el rotor, igualmente como vector, $v^e_{a,b,c}$ hace referencia a cada voltaje de las fases a, b, c en el estator y por último $v^r_{a,b,c}$ indica los diferentes voltajes de cada fase a, b, c del rotor. Por otro lado, la corriente es denotada por la Ecuación (3).

$$[i] = \begin{bmatrix} [i_e] \\ [i_r] \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} i_a^e & i_b^e & i_c^e \end{bmatrix}^t \\ \begin{bmatrix} i_a^r & i_b^r & i_c^r \end{bmatrix}^t \end{bmatrix} \tag{3}$$

donde i es el vector de corriente, i_e es la corriente de corriente en el estator, i_r es vector de corriente en el rotor, $i^e_{a,b,c}$ y $i^r_{a,b,c}$ denotan las corrientes de cada bovina del rotor y estator. Finalmente, el flujo es descrito por la Ecuación (4).

$$[\lambda] = \begin{bmatrix} [\lambda_e] \\ [\lambda_r] \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} \lambda_a^e & \lambda_b^e & \lambda_c^e \end{bmatrix}^t \\ \begin{bmatrix} \lambda_a^r & \lambda_b^r & \lambda_c^r \end{bmatrix}^t \end{bmatrix} \tag{4}$$

en este caso λ indica el vector de flujo, λ_e es el vector de flujo en el estator, λ_r es el vector de flujo en el rotor, $\lambda^e_{a,b,c}$ denota los flujos de cada fase a, b, c del estator y $\lambda^r_{a,b,c}$ denota los flujos de cada fase a, b, c del rotor.

Por otro lado, la ecuación mecánica que describe el comportamiento de la máquina de inducción se define por la segunda Ley de Newton, según la Ecuación (5):

$$T_e - T_m = J\ddot{\theta} \tag{5}$$

donde T_m es el par mecánico de carga, T_e es el par eléctrico, y J es la inercia del eje del motor. A partir del principio de trabajos virtuales se puede obtener el par eléctrico del motor (Aller, 2006):

$$T_e = \frac{\partial W_c'}{\partial \theta} = \frac{1}{2}[i]^t[\tau(\theta)]^t[i] \tag{6}$$

La expresión mostrada en (7) junto con Ecuación (1) llega a representar de forma matemática cómo se comporta la máquina de inducción en coordenadas primitivas. La Ecuación (7) es generada a partir de las Ecuaciones (6) y (5). La representación se observa a continuación.

$$\frac{1}{2}[i]^t[\tau(\theta)]^t[i] - T_m = J\ddot{\theta} \tag{7}$$

En estado estable, la Ecuación (5) es igual 0 y el par eléctrico se aproxima al par de carga.

5.2.2. Estimación par eléctrico

Una forma práctica de estimar el par eléctrico es usando los vectores tensión y corriente planteados como vectores espaciales. La transformación a vectores espaciales para la corriente y voltaje son establecidos con las Ecuaciones (8) y (9).

$$\vec{i_e} = \sqrt{\frac{2}{3}}\left(i_a + i_b e^{j\frac{2\pi}{3}} + i_c e^{j\frac{4\pi}{3}}\right) = \sqrt{\frac{2}{3}}\left(i_a + i_b\alpha + i_c\alpha^2\right) \tag{8}$$

$$\vec{v_e} = \sqrt{\frac{2}{3}}\left(v_a + v_b e^{j\frac{2\pi}{3}} + v_c e^{j\frac{4\pi}{3}}\right) = \sqrt{\frac{2}{3}}\left(v_a + v_b\alpha + v_c\alpha^2\right) \tag{9}$$

El procedimiento para calcular el vector espacial del flujo del estator se realiza mediante la Ecuación (10) con las variables medidas y transformadas a vectores espaciales. El valor de la resistencia del estator se puede medir mediante el acceso a los terminales de las bobinas del motor.

$$\vec{\lambda}_e(t) = \int_0^t \left(\vec{v}_e(\tau) - R_e\vec{i}_e(\tau)\right) d\tau \tag{10}$$

Posteriormente, se calcula el par eléctrico propuesto por Rengifo, Aller, Bueno, Viola, y Restrepo (2012), que se presenta en la Ecuación (11):

$$T_e = n_p(\vec{\lambda}_e \times \vec{i}_e) \tag{11}$$

5.2.3. Par de carga

El par de carga abarca la suma del par de fricción mecánico T_f interno del motor dado por $T_f = k_{loss}\omega_m$ y el par del freno magnético T_{fm}, presente en el sistema mecánico. Debido

a que el par de fricción puede expresarse como una función de velocidad, este llega a ser constante cuando se encuentra en régimen estacionario y, en cambio, para el par del freno este es constante todo el tiempo. También se debe considerar las perdidas por fricción T_{fc} y componentes frecuenciales de par relacionadas en una caja de engranajes (Kia, Henao, y Capolino, 2009) y de esta manera la Ecuación (7) debe extenderse con la Ecuación (12):

$$T_m = k_{loss}\omega_m + T_{fm} + T_{fc} + T_{gb}(t) \tag{12}$$

El par T_{gb} depende de las características de la caja de engranajes, para este estudio se utilizó una caja de engranajes de una sola etapa. Estas propiedades son importantes puesto el par de carga se debe considerar en el desarrollo del modelo. En la Figura 5 se representa de manera simplificada el sistema electromecánico y el modelo dinámico para análisis del par T_{gb} mencionado.

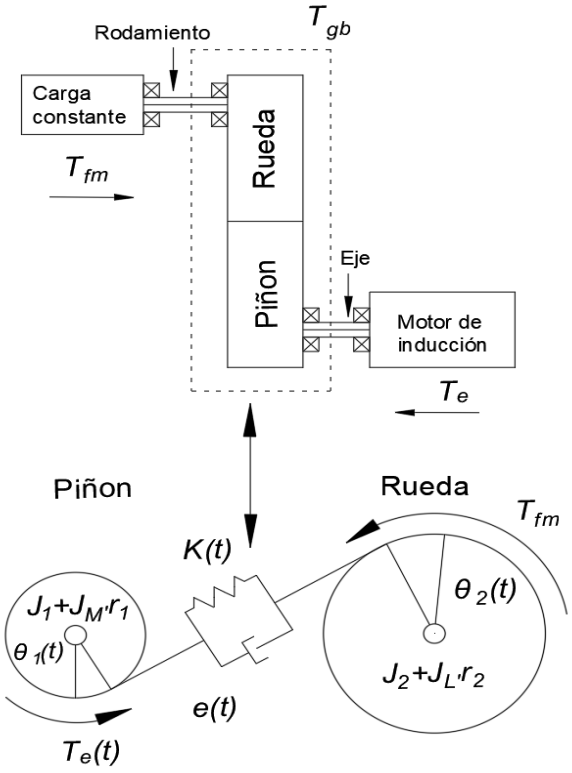


Figura 5

Representación del sistema electromecánico y modelo dinámico de la caja de engranajes (Kia y cols., 2009).

Por otro lado, en la Ecuación (13) se representa el modelo de la caja engranajes que genera T_{gb} (Kia y cols., 2009).

$$(J_1 + J_M)\ddot{\theta}_1(t) + r_1 K(t)\left[r_1\theta_1(t) + r_2\theta_2(t) + e(t)\right] + r_1 d_z\left[r_1\dot{\theta}_1(t) + r_2\dot{\theta}_2(t) + \dot{e}(t)\right] = T_e(t)$$

$$(J_2 + J_L)\ddot{\theta}_2(t) + r_2 K(t)\left[r_1\theta_1(t) + r_2\theta_2(t) + e(t)\right] + r_2 d_z\left[r_1\dot{\theta}_1(t) + r_2\dot{\theta}_2(t) + \dot{e}(t)\right] = T_L(t) \quad (13)$$

Donde:

J_1	es la inercia del piñón
J_2	es la inercia de la rueda
J_M	es la inercia del rotor de la máquina de inducción
J_L	es la inercia de la carga
r_1, r_2	es el radio del piñón y rueda respectivamente
d_z	es el coeficiente de amortiguamiento del punto de contacto
$K(t)$	es la función de rigidez del punto de contacto
$e(t)$	es la función del error en la transmisión
$T_e(t)$	es el par electromecánico
$T_L(t)$	es el par de carga
$\theta_1(t)$	es el ángulo rotacional del piñón
$\theta_2(t)$	es el ángulo rotacional de la rueda

De acuerdo con Kia y cols. (2009), la caja de engranes genera oscilaciones de par que podrían se puede representar como $T_{gb} = T_{rd} + T_{rA} + T_{rB}$ de forma que:

$$T_{rd} = A_{rd} sin(\omega_{rd} t + \phi_{rd})$$

$$T_{rA} = A_{rA} sin(\omega_{rA} t + \phi_{rA})$$

$$T_{rB} = A_{rB} sin(\omega_{rB} t + \phi_{rB})$$

donde, ω_{rd} es la frecuencia de engranaje, ω_{rA} es la frecuencia del piñón y por último, ω_{rB} es la frecuencia de la rueda. Este par T_{gb} permite modular la corriente del estator, y consecuentemente afecta el par eléctrico T_e.

Con estas descripciones matemáticas, se justifica que las variaciones en el par de carga, por ejemplo, aquellas inducidas desde la caja de engranajes debido a fallos, los caules causarían cambios en el par eléctrico dependiendo del nivel de severidad de diente roto y de ahí el interés

de usar esta señal para monitorear la condición de la caja de engranajes.

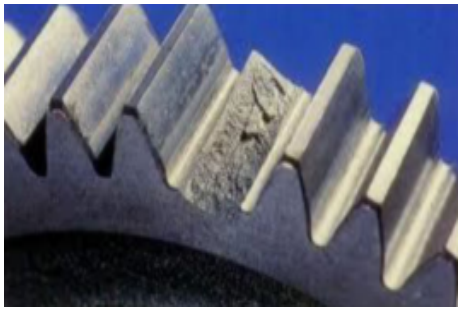
5.3. Engranes y caja de engranes

Los engranajes son los componentes de mayor importancia del sistema de transmisión. Es posible catalogarlos en engranajes de ejes paralelos, intersección de ejes, sin intersección de ejes y espaciales. Los engranes de ejes paralelos son conectados con ejes paralelos y transfieren potencia de forma muy eficiente. Los engranajes rectos y helicoidales son los dos principales tipos (R y cols., 2006). La caja de engranajes es un dispositivo mecánico usado con el fin de aumentar el par de salida o modificar la velocidad (RPM) del motor. El eje del motor, que está conectado al extremo de la caja y también mediante una determinada configuración interna de los engranajes, generan un par de salida y una velocidad que depende de la relación de transmisión(Babu, Reddy, Naresh, y Reddy, 2016).

En lo que se refiere a los **fallos producidos en los engranajes**, estos ocurren por lo general en sus dientes y algunas de las causas son una mala lubricación o alineación, así como la fatiga, velocidades muy altas, entre otros. Los diferentes fallos producidos en los dientes de una caja de engranes incluyen:la generación de grietas (crack), rayadura (scuffing), picadura (pitting) y dientes de los engranajes rotos (broken tooth). Lo más perjudicial para la maquinaria en aspectos generales es el fallo por diente roto. Este fallo se produce por una grieta ubicada en la raíz del diente y suele extenderse hacia el diente o también, parte del mismo se rompe. El nivel de severidad de rotura del diente es determinado mediante el porcentaje de volumen perdido (Rodríguez y cols., 2014).

Figura 6

Engranaje recto con rotura de diente de condición severa.



Nota: *Se presenta un engranje con diente completamente roto (Llivicura Orellana, 2019).*

En la Tabla 1 se presenta una tabla realizada por Llivicura Orellana (2019), en donde se indica el nivel de diente roto, las razones por las cuales se genera y aspectos relevantes del diente. Como se puede observar, la severidad de fallo aumenta poco a poco hasta que el diente se desprende completamente.

Tabla 1

Tabla de nivel de severidad de diente roto.

Nivel de severidad	Causa de fallo	Estado del diente	Porcentaje de volumen de rotura (%dr)
Fallo leve	Carga excesiva de trabajo por un largo periodo	Pequeñas grietas en la raíz del diente provocan desprendimiento de un pequeño porcentaje del diente.	%dr <=25
Fallo moderado	Carga excesiva de trabajo por un largo periodo de un engranaje con fallo leve de rotura.	Desprendimiento de más de la mitad del diente.	25 <%dr<=50
Fallo severo	Carga excesiva de trabajo por un largo periodo de un engranaje con fallo moderado de rotura.	El diente se rompe desde la raíz completamente.	%dr>50

Nota: *Tabla de diferentes porcentajes de niveles de fallo de diente roto (Chingal Imaicela, 2018).*

5.4. Adquisición del par eléctrico - Plan experimental

Para el trabajo se midieron señales de corriente y voltaje de un banco de pruebas con una caja de engranajes rectos ubicada en el Laboratorio de Vibraciones de la Universidad Politécnica Salesiana, Cuenca, Azuay, Ecuador. Según un plan experimental que está disponible para revisión en el grupo de investigación, dependiendo de la severidad del fallo de diente roto, se presentan diferentes patrones que hacen posible identificar y diferenciar el estado del equipo.

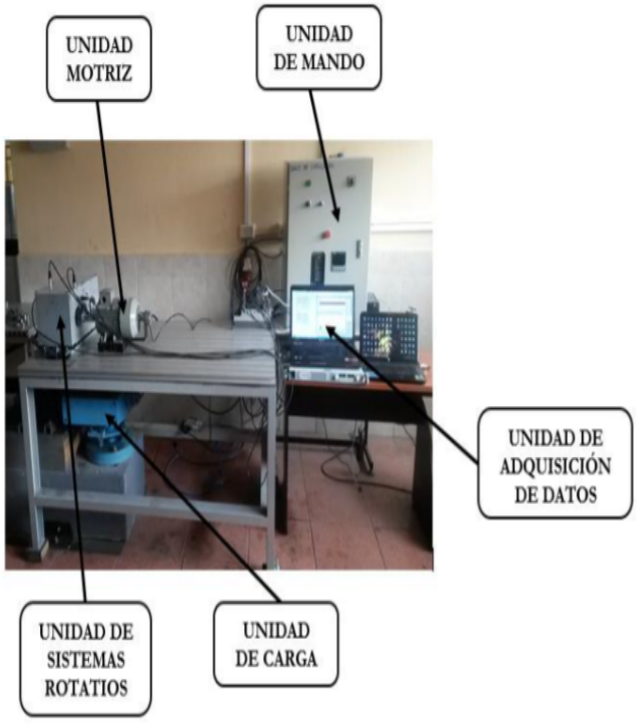
En un banco de vibraciones con condiciones reales y controladas, está acoplada caja de engranajes que permite generar bases de datos del sistema electromecánico. Los datos que se

suelen medir y registrar son de señales de corriente, tensión, velocidad de entrada y salida, vibración, acústica y emisión acústica. Para este proyecto se estableció el levantamiento de datos de las señales a velocidad constante del motor. Los datos mencionados se adquirieron por programas desarrollados en LabVIEW y MATLAB.

Por otro lado, en cuanto al banco de vibraciones de la Universidad Politécnica Salesiana, se debe mencionar que esta tiene el equipamiento adecuado para la simulación de desperfectos presentes en la industria y maquinaria en general, como desbalanceo, desalineación y combinación de fallos de rodamiento y engranes de maquinaria rotativa. El banco consta de cinco unidades principales: Unidad de mando, Unidad motriz, Unidad de carga, Unidad de sistemas rotativos y Unidad de adquisición de datos. En la Figura 7 se muestra la disposición de la unidad que conforman en el banco.

Figura 7

Banco de prueba de la Universidad Politécnica Salesiana, Sede Cuenca



Nota: *Partes del banco de prueba del Laboratorio de Vibraciones de la Universidad Politécnica Salesiana, Sede Cuenca (Loaiza Sánchez, 2021).*

Para generar una base de datos de severidad de fallo de engranajes rectos a velocidad

continua, en el banco de pruebas se dispuso de un motor, una caja de engranajes de una etapa, una polea con su correa trapezoidal y freno magnético. Para medir los diferentes fenómenos físicos se usaron un encoder lasér, un tacómetro, cuatro acelerómetros, dos sensores de emisión acústica, tres pinzas amperimétricas, seis sensores de voltaje y dos micrófonos de condensador conectados a una computadora portátil mediante un chasis que adquiere datos (DAQ,NIcDAQ) con la ayuda de un programa desarrollado por miembros del grupo de investigación en LabVIEW. Dentro de este plan, las señales medidas de los diferentes fallos son tomadas tanto de la caja de engranajes y del motor.

Las propiedades del equipo mecánico usado y la caja de engranajes se describen en las Tablas 2 y 3 respectivamente. Las propiedades del motor utilizado con el fin de generar esta base de datos se observa en la Tabla 4, el cual ha sido conectado de la forma doble estrella paralelo (220 V).

Tabla 2

Características de la caja de engranajes del plan experimental provisto por el grupo GIDTEC

Especificación del equipo mecánico	
Caja de engranes	Una fase con engranajes rectos
Rodamientos:	NTN 6005 Z2C3
Tipo de correa:	Correa trapezoidal
Tipo de lubricación:	Baño de aceite
Aceite:	Gulf HARMONY AW ISO VG 68
Cantidad de aceite:	1.8 gal

Nota: *Carácteristicas del equipo mecánico del banco de pruebas obtenido del plan expermiental realizado por el grupo de GIDTEC (Ortega Lucero, 2021).*

La caja de engranes rectos posee el tipo de fallo de diente roto. En la Tabla 3 se puede observar las características de los engranajes como se mencionan anteriormente. Los fallos se incorporan en un solo diente del piñón (Z1). El estado de salud del engrane ira empeorando al incorporar el fallo en un nivel leve, moderado y severo. Para determinar la severidad del fallo se utiliza el porcentaje de volumen de perdida del diente. De este modo, los fallos generados comprenderán una perdida uniforme del 12.5 % al 100 % de su volumen, que inicia desde las esquinas. En la Figura 8 se puede observar un engrane con fallo con una perdida del 25 % de su diente. De esta forma, se tienen disponibles 8 condiciones que representa diferentes niveles de fallo del diente roto (P2-P9) y una condición en estado saludable del engrane (P1). En relación con las partes del banco de pruebas del laboratorio y la ubicación de los sensores se

presenta en el diagrama de la Figura 9 mientras la Figura 10 muestra su configuración con los respectivos módulos dela NI Compact DAQ para los sensores.

Tabla 3

Características de la caja de engranajes del plan experimental "Severidad de fallo de engranaje recto a velocidad constante"

Características de los engranes	
Tipo de engranajes:	Rectos
Número de etapas:	1
Material:	Acero E410
Modulo:	2.25
Ángulo de presión:	20 grados
Cantidad de aceite:	1.8 gal
Transmisión de relación cercana:	0.67
Ancho:	20.7 mm
Número de dientes Z1/Z2:	32/ 48

Nota: *Carácteristicas del los engranes de la caja del banco de pruebas obtenido del plan expermiental realizado por el grupo de GIDTEC (Ortega Lucero, 2021).*

Tabla 4

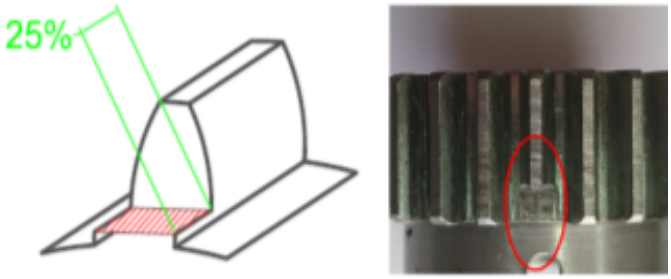
Resumen de las características del motor

Características del motor	
Motor	Motor A
Marca:	Siemens
Denominación:	1LA7096-6YA60
Tipo de motor:	Trifásico
Potencia nominal:	2Hp
Conexiones:	YY/Y
Tensión nominal:	220/440 V
Corriente nominal:	7.8/3.9
Cos:	0.77
Número de pares polos:	3
Velocidad nominal:	1110 rpm
Clases de aislamiento:	Clase F
Momento de inercia:	$0,0035 kg.m^2$

Nota: *Carácteristicas del motor del banco de pruebas obtenido del plan expermiental realizado por el grupo de GIDTEC (Ortega Lucero, 2021).*

Figura 8

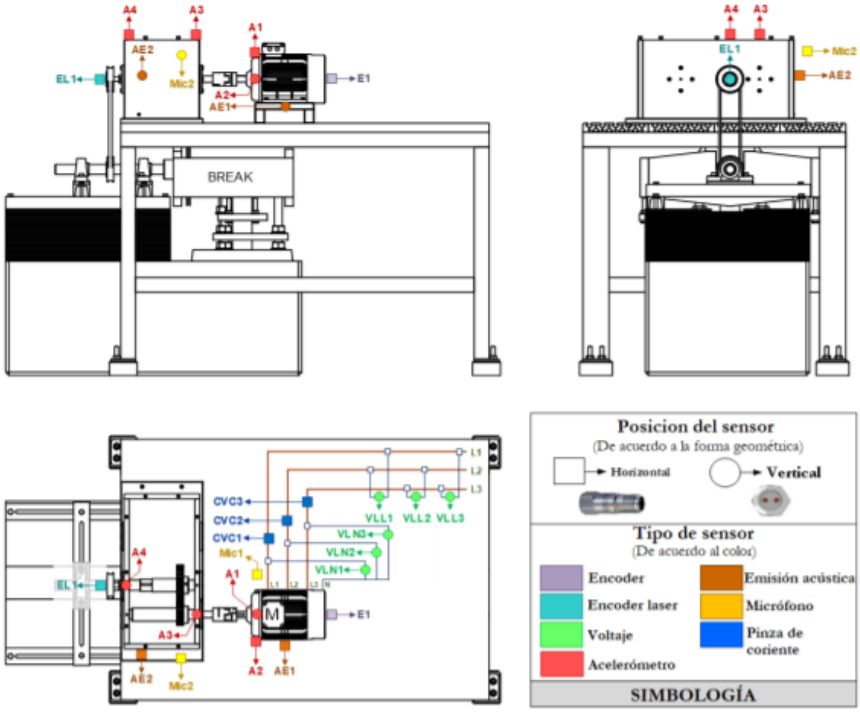
Engranaje recto con perdida de 25 % de su diente, fallo implementado físicamente



Nota: *Ejemplo del 25 % de perdida de volumen de diente de engranajes (Llivicura Orellana, 2019).*

Figura 9

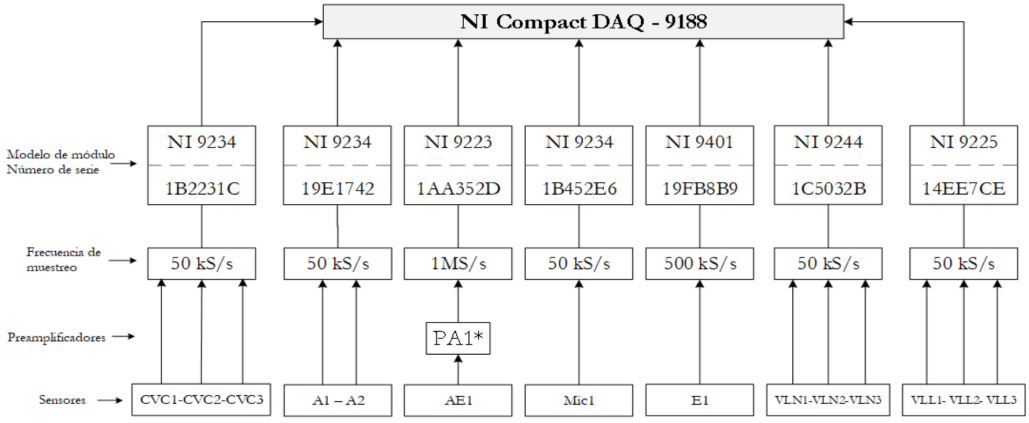
Disposición de los elementos mecánicos del experimento



Nota: *Ubicación de los diferentes sensores en el banco de preubas obtenido de Plan experimental de grupo GIDTEC (Ortega Lucero, 2021).*

Figura 10

Configuración de sensores y módulos del NI Compact DAQ 9188



Nota: *Unicacion de sensores del Chasis NI Compact DAQ 9188 btenido de Plan experimental de grupo GIDTEC (Ortega Lucero, 2021).*

Las señales son medidas durante 10 segundos y también se realizan 10 repeticiones para todas la pruebas. En este caso se utilizó el motor M1 a velocidad constante que se conecta a la red eléctrica con un interruptor mecánico. En relación con las cargas para las pruebas, se ha introducido tensiones de alimentación del freno magnético de 0V, 10V y 20V que representan las cargas L1, L2 Y L3 respectivamente. Las condiciones del engranaje se nombran P1 cuando el diente está en estado saludable y hasta P9, donde el diente se ha roto de completamente. En la Figura 11 se muestra las combinaciones mencionadas de niveles fallos y cargas.

Figura 11

Combinación para la adquisición de la base de datos

Canales	Reptación	Motor	Carga	Códigos de falla
Todos	R1	M1	L1	P1
	R2		L2	P2
	R3		L3	P3
	R4			P4
	R5			P5
	R6			P6
	R7			P7
	R8			P8
	R9			P9
	R10			

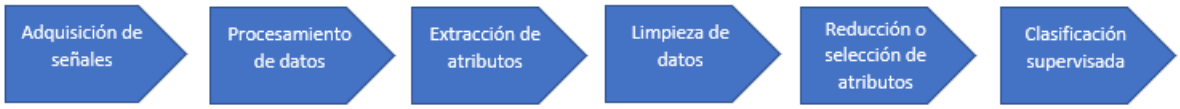
Nota: *Configuración y combinaciones para la dquicisión de las señales de la base de datos obtenido de Plan experimental de grupo GIDTEC (Ortega Lucero, 2021).*

5.5. Clasificación de fallos basado en datos

La Figura 12 indica de forma general las etapas de la aplicación del aprendizaje automático, las cuales son descritas en los siguientes párrafos cuando se trata del caso de aplicación en la detección y diagnóstico de fallos.

Figura 12

Esquema del proceso para clasificación de fallos basado en datos.



Nota: *Pasos seguidos para llevar a cabo el proceso para clasificación de fallos basado en datos Loaiza Sánchez (2021).*

En primer lugar, para la etapa de **adquisición de la señal** hay varios fenómenos físicos ocurriendo dentro un determinado sistema y de este modo resulta conveniente medir sus variables físicas como señales de corriente, acústicas, voltaje, vibratorias y otras más que se pueden adquirir y almacenar mediante diferentes instrumentos. Por otro lado, es necesario conocer las condiciones normales y de fallo de la máquina, escoger los sensores apropiados y establecer la ubicación de los mismos para tener una señal adecuada y válida (Loaiza Sánchez, 2021). En este proyecto se utilizó la base de datos de señales del par eléctrico del motor trifásico disponible en el grupo GIDTEC, la cual está organizada según las condiciones de funcionamiento de una caja de engranajes rectos emplazada en un banco de pruebas.

Después se encuentra la etapa de **procesamiento de la señal**, en la cual se debe realizar las modificaciones necesarias para que los datos de la señal sean representados en un determinado dominio. Por lo general, al medir señales se utiliza el dominio del tiempo (series temporales) pero también se suele usar dominios como la frecuencia, usando la Trasformada Rápida de Fourier, tiempo-frecuencia, y otros más (Feng, Liang, y Chu, 2013). En el caso de este trabajo, se usaron los mapas de Poincaré como una representación 2D de la serie temporal de la señal del par eléctrico para extraer atributos o características.

Al realizar la **extracción de atributos**, generalmente se calculan indicadores estadísticos en el dominio de representación apropiado según el trabajo a realizar. Con relación a los

dominios del tiempo y de la frecuencia se obtienen los valores de indicadores como la desviación estándar, la media, la kurtosis y muchos más (Sánchez Loja, 2018). En el proyecto presente, puesto que se trabaja con los mapas de Poincaré, se obtienen los 11 atributos de Poincaré presentados con mayor detalle en la sección 5.6.

Seguidamente, para la etapa de **limpieza de datos** se realiza un análisis exploratorio de datos con el fin de identificar valores atípicos, que son descartados para las etapas posteriores. Estos valores se generan en situaciones no controladas en el proceso de adquisición de señales. Debido a que los resultados de los indicadores estadísticos tienen rangos de valores máximos y mínimos diferentes, se realiza un proceso de normalización o estandarización del dominio de atributos para homogeneizar las magnitudes de los indicadores (Sánchez Loja, 2018). Una vez mencionado esto, para el trabajo presente se normalizaron los valores obtenidos para cada atributo sobre el intervalo [0-1].

La etapa de **reducción o selección de atributos** consiste en deshacerse de atributos redundantes o irrelevantes que no aportan para diferenciar entre condición normal y de fallo dentro del sistema. Por otro lado, se suele reducir la dimensión del vector de indicadores con técnicas de mapeo (Sharma y Saroha, 2015), no obstante; al realizar esta reducción, se generan atributos artificiales que no se relacionan con el sistema físico estudiado y son difíciles de interpretar. En caso de requerir de la reducción y clasificación de atributos importantes, la minería de datos posee herramientas adecuadas para realizarlo, porque ha demostrado ser muy eficaz en el manejo de información compleja. Aplicando esto al trabajo presente, se debe mencionar que, para la selección de indicadores, se realizó un *ranking* de los atributos más importantes a menos importantes basado en métricas como: Chi Cuadrado (X^2), CDbw, Random Forest y ReliefF.

Por último, en la etapa de **clasificación** es posible reconocer patrones de los datos de la señal estudiada a partir de la extracción de características y un entrenamiento adecuado del modelo. Al conocer cada uno de los diferentes niveles de severidad de fallo en el sistema se puede realizar un mapeo de todos los datos. Durante esta fase, el aprendizaje automático posee una variedad de modelos y algoritmos para clasificar los datos según su nivel de falla, es decir, permite desarrollar sistemas de diagnóstico de fallos. Los modelos de aprendizaje automáticos utilizados en este trabajo son: KNN y RF.

5.6. Diagrama de Poincaré

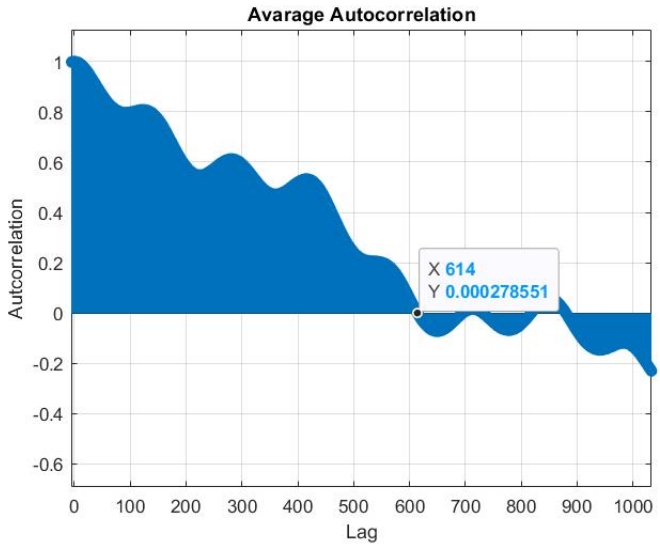
Una de las herramientas típicas para analizar sistemas dinámicos no lineales y caóticos es el diagrama de espacio de fase o el diagrama de espacio de fase, el cual es una gráfica donde los estados de un sistema se representan como una función del tiempo. En sistemas mecánicos, el diagrama de espacio de fase se puede obtener trazando trayectorias 2-D con respecto al desplazamiento y la velocidad; sin embargo, el diagrama de Poincaré (DP) es un diagrama para cuantificar la auto-semejanza de series temporales con el fin de descubrir comportamientos caóticos. Por otro lado, el DP es considera un desfase arbitrario entre dos muestras; esto tiene la ventaja de considerar un sistema embebido de retardo de tiempo, lo cual es útil para modelar series de tiempo caóticas (Cerrada y cols., 2020).

El diagrama de Poincaré se puede graficar al trazar una serie temporal $x(t)$ de alguna variable medida respecto a sí misma, pero desplazada $x(t+\tau)$ (Alligood, Sauer, y Yorke, 1996). Al seleccionar correctamente un desplazamiento τ, o también llamado lag, el cual es un valor obtenido cuando la autocorrelación de $x(t+\tau)$ es cero o muy cercana a cero. Mediante esta representación gráfica, se pueden calcular atributos o características para clasificar fallas y en este caso específico para clasificar grado de severidad de rotura de diente. El gráfico de Poincaré genera un clúster bidimensional $x(t)$ vs $(x(t+\tau)$. Un clúster hace referencia a datos agrupados en el espacio. Por otro lado, el retardo τ podría escogerse de forma arbitraria para el caso de una cantidad infinita de datos sin ruido, pero cuando se trata de datos finitos y ruidosos, el desfase debe escogerse de forma que las muestras en $x(t)$ no estén correlacionadas entre sí y esto se logra identificando el cruce por cero en el gráfico de autocorrelación (Cerrada y cols., 2020).

En las Figuras 13 y 14 se observa un gráfico de la autocorrelación promedio y un diagrama de Poincaré respectivamente. El valor de autocorrelación varía de -1 a 1. Un valor entre -1 y 0 representa una autocorrelación negativa y, por otro lado, un valor entre 0 y 1 representa una autocorrelación positiva. En la Figura 13 se puede ver que el valor del retraso o *lag* es de 614 debido a que la autocorrelación es muy cercana a 0. La autocorrelación mostrada tiene el nivel de severidad P1, es decir, la caja de engranajes se encuentra en estado saludable. Este proceso se realizó para todos los niveles de severidad de diente roto, desde P1 hasta P9. El valor del retraso fue determinado en el trabajo de Mejía (2022) y a partir del mismo, se tomaron tanto los diagramas de Poincaré requeridos para todas las señales y se calcularon los diferentes atributos de Poincaré.

Figura 13

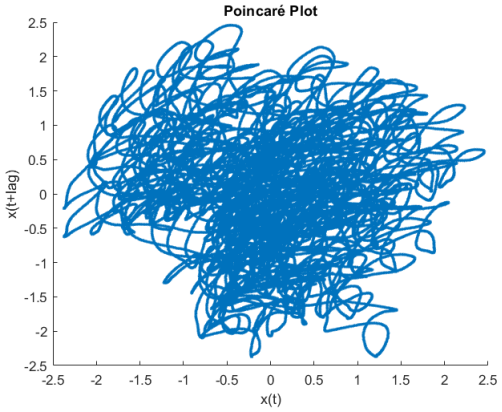
Gráfico de Autocorrelación promedio para el nivel de severidad de diente roto P1.



Nota: *En la grafica de autocorelación promedio para el nivel de severidad de diente roto P1 se obtuvo un lag de 614 (Mejía, 2022).*

Figura 14

Ejemplo de diagrama de Poincaré para el nivel de severidad de diente roto P1.



Nota: *A partir del diagrama de Poincaré generado mediante valor de autocorelación se obtienen los atributos de Poincaré. La imagen se obtuvo de (Mejía, 2022).*

A continuación se presentan los **atributos de Poincaré**, los cuales pueden ser obtenidos mediante el diagrama de Poincaré (DP). Este también puede ser analizado para resaltar su

asimetría significativa (Loaiza Sánchez, 2021). Se ha estudiado la capacidad del DP para producir grupos de puntos bien separados y así diagnosticar el estado actual de maquinaria rotativa. En esta sección se han indicado 11 características de Poincaré obtenidas del diagrama o gráfico de Poincaré donde $x(t)$ representa la serie temporal analizada, la cual en este trabajo es la señal de par eléctrico. Los atributos usados aquí fueron obtenidos del artículo de Peña, Cerrada, Medina, Cabrera, y Sánchez (2022) y se han transcrito continuación:

1.- **Dispersión de puntos SD1:** El SD1 mide la dispersión de puntos (desviación estándar) que son perpendiculares al eje de la línea de identidad $y = x$. Entonces, SD1 se calcula mediante la Ecuación (14):

$$SD1^2 = \frac{1}{2}\left\{x - x(t+\tau)\right\} = \frac{1}{2}SDSD^2 \tag{14}$$

2.- **Dispersión de puntos SD2:** El $SD2$ es la medida de la dispersión de puntos (desviación estándar) con respecto al eje $y = -x$, luego $SD2$ se calcula mediante la Ecuación (15):

$$SD2^2 = 2var\left\{x(t)^2\right\} - \frac{1}{2}SDSD^2 \tag{15}$$

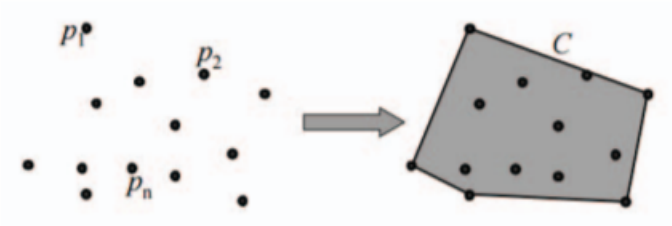
3.- **Relación SD:** Esta característica mide la relación cuantitativa entre SD1 y SD2, como se muestra en la Ecuación (16):

$$SD2 = \frac{SD1}{SD2} \tag{16}$$

4.- **Zona de casco convexo:** Dados los puntos de ajuste $P = P_i$, el casco convexo (convex hull) es el conjunto convexo más pequeño C, tal que $P \subset C$, ver en la Figura 15. Existen diferentes algoritmos para establecer los vértices convexos del casco. Dentro de este proyecto interno, el algoritmo propuesto en Barber, Dobkin, y Huhdanpaa (1996) se utiliza para calcular el casco convexo del diagrama de Poincaré. El casco convexo C genera un polígono convexo y al calcular su área, este valor se considera un atributo.

Figura 15

Casco convexo y área del casco convexo.



Nota: *Ejemplo de desarollo de Convex Hull (Medina y cols., 2017).*

5.- **Eje centrado SDC:**

Esta característica es calculada por Ecuación (17):

$$SDC = \frac{\sqrt{\frac{1}{2}\sum_{i=1}^{n}(d_i^1)^2}}{\sqrt{\frac{1}{2}\sum_{i=1}^{n}(d_i^2)^2}} \tag{17}$$

Donde d_i^1 y d_i^2 se calculan mediante las Ecuaciones (18) y (19) siendo $\overline{x_i}$ y $\overline{y_i}$ valores medios sobre los datos disponibles:

$$d_i^1 = \frac{\|(x_i - \overline{x_i}) - (y_i - \overline{y_i})\|}{\sqrt{2}} \tag{18}$$

$$d_i^2 = \frac{\|(x_i - \overline{x_i}) - (y_i - \overline{y_i})\|}{\sqrt{2}} \tag{19}$$

6.- **Dispersión de puntos SD1C:** Esta es una medida de la dispersión puntual del diagrama de Poincaré que se obtiene calculando la varianza de la distancia entre los puntos y la línea $y = x$ centrada alrededor del centroide del gráfico con la Ecuación (20):

$$SD1C = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i^1)^2} \tag{20}$$

7.- **Dispersión de puntos SD2C:** Es una medida de la dispersión puntual del diagrama de Poincaré donde se calcula la varianza de la distancia entre los puntos y la línea $y = x$ centrada alrededor del centroide del gráfico, de la forma a continuación:

$$SD2C = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i^2)^2} \tag{21}$$

8.- **Dispersión de puntos $SD1C_{up}$:** Esta característica está relacionada con la distancia D_{up}^i entre los puntos del diagrama de Poincaré ubicados sobre la línea de identidad $y = x$ y esta línea de identidad. Esto se calcula mediante la Ecuación (22):

$$SD1_{up} = \sqrt{\frac{1}{n_{up}}\sum_{i=1}^{n_{up}}(D_{up}^i)^2} \tag{22}$$

Donde n_{up} es el número total de puntos sobre la línea $y = x$.

9.- **Dispersión de puntos $SD1C_{down}$:** Esta característica está relacionada con la distancia D_{down}^i entre los puntos del diagrama de Poincaré ubicados debajo de la línea de identidad, $y = x$ esta línea de identidad. Esto se calcula mediante la Ecuación (23):

$$SD1_{down} = \sqrt{\frac{1}{n_{down}}\sum_{i=1}^{n_{down}}(D_{down}^i)^2} \tag{23}$$

Donde n_{down} es el número total de puntos sobre la línea $y = x$.

10.- **Relación de asimetría superior:** Este atributo es la medida de la relación entre la asimetría superior con respecto a la distancia media $SD1_A$ entre todos los puntos y la recta $y = x$. Esto se calcula como se muestra en la Ecuación (24):

$$C_{up} = \frac{SD1_{up}^2}{SD1_A^2} \tag{24}$$

11.- **Relación de asimetría inferior:** Esta característica es análoga a la característica de relación de asimetría superior, y se calcula mediante la Ecuación (25):

$$C_{down} = \frac{SD1_{down}^2}{SD1_A^2} \tag{25}$$

5.7. Selección de atributos

Los sistemas de predicción que suelen usar algoritmos de aprendizaje automático requieren de una implementación apropiada y es importante tener en cuenta los atributos usados para entrenar los modelos computacionales que realizan las predicciones. Para poder seleccionar los

atributos adecuados y más representativos, se trabajó con los atributos de Poincaré explicados en sección 5.6 del documento. Uno de los propósitos de seleccionar atributos es disminuir la dimensionalidad del conjunto de atributos a través de la selección del subconjunto de atributos para considerar aquellos de mejor desempeño bajo algún criterio de clasificación (H. Liu, Motoda, Setiono, y Zhao, 2010).

5.7.1. ReliefF

Es posible resumir que los algoritmos de ReleifF son estimadores de atributos generales y permiten conocer las dependencias condicionales entre atributos. Estos algoritmos se utilizan principalmente como un método de selección de subconjuntos de características (Robnik-Šikonja y Kononenko, 2003). El *Ranking* con ReliefF estima la calidad de los atributos que permiten discriminar entre instancias vecinas (Sánchez Loja, 2018). Además, permite evaluar calidad de los atributos capturando varias de sus muestras de forma aleatoria y luego calculando sus k vecinos más cercanos de cada una de las clases existentes (Kononenko, 1994), es decir, tanto la misma clase como las diferentes clases disponibles. Una vez realizado esto, se renueva los valores de un vector de ponderación W. Este vector otorga mayor peso a los atributos que discriminan notoriamente entre vecinos de otras clases. De este modo, al tomar los atributos que poseen un mayor peso se seleccionan los mejores atributos o en otras palabras los atributos más adecuados. W se define por la Ecuación (26):

$$W_f^{i+1} = W_j^i + \sum_{c=class(x)} \frac{\frac{p(x)}{1-p(class(x))} \sum_{j=1}^{k} d_f(x, M_j(x))}{m * k} - \sum_{j=1}^{k} \frac{d_f(x, M_j(x))}{m * k}, \tag{26}$$

donde W_f indica la importancia de cada atributo, puesto que se relaciona con su el peso de la característica f, d_f indica la distancia entre dos muestras de las características o los atributos f, $Hj(x)$ representa la cantidad de muestras vecinas desde el tipo de muestra de x, $Mj(x)$ son las muestras vecinas de otras clases y finalmente $p(x)$ indica la probabilidad de clase.

5.7.2. Composing Density Between and With clusters (CDbw)

Es un método de selección de atributos, se basa en la medición de la estructura de clústeres. La compacidad y la separación son ejemplos de características geométricas de los clústeres a los que el índice CDbw pone énfasis. Un clúster i se compone por n_i muestras $x \in R_n$ y, además del centroide (punto medio del clúster), hay r_i puntos representativos establecidos v_{iri} que son producidos en un procedimiento que se repetirá varias veces o iterativo (Peña y

cols., 2018). El punto que más lejos se encuentra del centroide es escogido como el punto de inicio representativo en la primera iteración. Después el punto representativo se escoge de forma que se ubique lo más separado del punto elegido antes, y así repetidas veces (S. Wu y Chow, 2004).

Al tener una partición de clústeres, en vez de solamente usar la información de distancia, se encuentra la información de densidad intra-clúster e inter-clúster que también es considerado por el índice CDbw, planteando así la Ecuación (27):

$$CDbw(c) = IntraD(c)Sep(c) \tag{27}$$

donde c es el número de clústeres dentro de la partición, $Sep(c)$ es usada para medir la separación entre clústeres y finalmente $IntraD(c)$ representa la densidad intra-clúster. En este contexto, $Sep(c)$ considera tanto las distancias inter-clúster como la densidad inter-clúster, como se propone en la Ecuación (28):

$$Sep(c) = \sum_{i=1}^{c} \sum_{\substack{j=-1, \\ i \neq j}}^{c} \frac{\|C\_R(i) - C\_R(j)\|}{1 + inter\_D(c)}, c > 1 \tag{28}$$

donde $C\_R(i)$ y $C\_R(j)$ son el par que más cerca se encuentra de representaciones de dos conglomerados vecinos i y j, e $Inter\_D(c)$ representa la densidad de inter-clúster la cual permite definir la densidad en las áreas inter-clúster. De este modo, la Ecuación (29) muestra:

$$Inter\_D(c) = \sum_{i=1}^{c} \sum_{\substack{j=-1, \\ i \neq j}}^{c} \frac{\|C\_R(i) - C\_R(j)\|}{\|sd(i) + sd(j)} d(u_{uij}, c > 1 \tag{29}$$

donde u_{ij} es el punto medio entre el par $C\_R(i)$, $C\_R(j)$, $sd(.)$ es el vector de desviación estándar de un conglomerado, y $d(u_{uij}) = \sum_{j=1}^{c} f(x_k, u_{ij})$ es una densidad, n_i y n_j son el número de muestras pertenecientes a los conglomerados i y j, respectivamente, y x_k es un punto de datos de los conglomerados. La función $f(x_k, u_{ij})$ se presenta en la Ecuación (30):

$$f(x_k, u_{ij}) = \begin{cases} 1 & \text{if } x_k, u_{ij} < \frac{\|sd(i), sd(j)\|}{c} \\ 0 & \text{Otherwise} \end{cases} \tag{30}$$

Por otra parte, $Intra\_D(c)$ representa la cantidad de puntos pertenecientes a la vecindad de puntos representativos de los conglomerados, tal como se propone en la Ecuación (31) a continuación:

$$Intra\_D(c) = \sum_{i=1}^{c} \sum_{j=1}^{r_i} Denv(v_{ij}), c > 1 \tag{31}$$

donde r_i es el número de puntos representativos del grupo i-ésimo, y $Den(v_{ij}) = \sum_{l=1}^{n_i} f(x_l, v_{ij})$ es una densidad, x_l es una muestra perteneciente al conglomerado i-ésimo, v_{ij} es el punto representativo j-ésimo del conglomerado i-ésimo; $f(x_l, v_{ij})$ viene dada por la Ecuación (32):

$$f(x_k, u_{ij}) = \begin{cases} 1 & \text{if } \|x_k, u_{ij}\| \leq sd_a \\ 0 & \text{Otherwise} \end{cases} \tag{32}$$

donde sd_a es el promedio de la desviación estándar de todos los vectores de desviación estándar sd asociados con cada grupo.

La densidad y la separación inter-clúster son considerablemente altas para los clústeres bien separados, luego se aplica esta métrica para evaluar la estructura del clúster obtenido cuando se selecciona un subconjunto de características significativas o atributos relevantes (Halkidi y Vazirgiannis, 2002). En este trabajo se usa una versión modificada de la Ecuación (27) que se puede revisar con mayor detalle en Peña y cols. (2018).

5.8. Algoritmos de aprendizaje automático

El aprendizaje automático es una rama de la inteligencia artificial. Las técnicas basadas en el aprendizaje automático han sido implementadas efectivamente de diferentes formas en actividades que consisten, por ejemplo, en: reconocer patrones, visión artificial, finanzas, el entrenamiento de modelos computacionales (para lo cual son aplicados en este trabajo) hasta implementaciones biomédicas y médicas (El Naqa y Murphy, 2015).

5.8.1. Random Forest

Son bosques de decisión aleatorios formados por un conjunto de árboles de decisión, los cuales se forman mediante un algoritmo que introduce una aleatoriedad para reducir la correlación entre los árboles (García Ruiz de León, 2018). Cuando el bosque aleatorio está construido, se lo utiliza para realizar la predicción de cada clase. De este modo, se puede definir que un bosque aleatorio es un clasificador que consta de una colección de clasificadores estructurados en árboles $\{h(x, \theta_k), k = 1, ...\}$ donde θ_k son vectores aleatorios independientes distribuidos de manera idéntica y cada árbol emite un voto unitario para la clase más popular

de la entrada x (Breiman, 2001). En cuanto a entrenar un clasificador basado en árboles, es similar a construir el árbol de forma recursiva.

En resumen, es posible decir que el algoritmo RF construye una gran cantidad de árboles de decisión (Yang, Park, y Kim, 2000) a partir de un subconjunto de datos de un conjunto o base de datos de entrenamiento mediante el uso del empaquetado (en inglés *bagging*), que es un meta-algoritmo para mejorar la clasificación, y modelos de regresión de acuerdo con la estabilidad y precisión de clasificación. El *bagging* reduce la variación y ayuda a evitar un ajuste excesivo síncronamente. La idea es crear varios subconjuntos de datos a partir de una muestra de entrenamiento elegida al azar con reemplazo. Cada nueva colección de subconjuntos generada a partir del remplazo de datos se usa para entrenar árboles de decisión. Cada clasificador de árbol se denomina predictor de componentes. El algoritmo toma decisiones contando los votos de los componentes predictores de cada clase y luego seleccionando la clase ganadora en términos del número de votos (Breiman, 2006).

RF genera varios árboles utilizando la metodología CART (árboles de clasificación y regresión) hasta el tamaño máximo y sin poda de los árboles. CART aumenta la clasificación y la regresión árboles para predecir variables dependientes continuas (regresión) y variables predictoras categóricas (clasificación) (Breiman, Friedman, Olshen, y Stone, 1984). Hay algunos pasos básicos en la metodología CART, los cuales según Breiman y cols. (1984) son:

1. Construcción de árboles de decisión: El proceso de construcción del árbol comienza con separar el nodo raíz en nodos binarios mediante una pregunta muy simple de la forma si x es mayor a d, que es $x \leqslant d$? Aquí, x son variables en el conjunto de datos y d es un número real. Inicialmente, todas las observaciones se ubican en el nodo raíz. CART implementa un algoritmo intensivo en computadora que busca la mejor división en todos los puntos de división posibles para cada variable.

2. Paro de construcción de árboles de decisión: CART detiene el proceso de división cuando:

   - Hay solo una observación en cada uno de los nodos secundarios.
   - Todas las observaciones dentro de cada nodo hijo tienen la distribución idéntica de variables predictoras, lo que hace imposible la división.
   - El usuario establece un límite externo en el número de niveles en el árbol máximo previamente.

Otros de los procedimientos aleatorios que RF realiza tienen que ver con extraer una cantidad de muestras de un conjunto de entrenamiento al azar (Breiman, 1996). Cada clasificador base en el conjunto se entrena en un *bootstrap* de la totalidad de los datos disponibles. Sin embargo, cada una de estas réplicas de *bootstrap* tiende a omitir aproximadamente un tercio de las muestras. En consecuencia, cada elemento en la muestra de tamaño n entrena aproximadamente $(2/3)k$ de todos los clasificadores en el conjunto, de modo, que, en promedio, alrededor de un tercio está fuera de la bolsa que en inglés sería *Out-Of-Bag* (OOB) y estos datos pueden usarse para validar los $k/3$ restantes clasificadores donde n es el número de datos de entrenamiento, k es el número total de clasificadores de un solo árbol (Yang, Di, y Han, 2008).

Después del procesamiento del remuestreo o en inglés *bootstrap*, el otro procedimiento aleatorio aparece en la división de nodos durante la construcción del clasificador de árbol (Breiman y cols., 1984). A diferencia del algoritmo normal de división del árbol de decisiones similar a CART. En cuanto a CART dentro del algoritmo de RF busca solo en n variables cuáles tienen un número pequeño y extraído al azar de todas las variables M en lugar de variables completas. Para este algoritmo, cualquiera que sea el procesamiento de *bootstrap* o de selección aleatoria de variables para dividir el nodo, ambos marcan la diferencia en árboles y bosques. Por lo tanto, estas dos fuentes de aleatoriedad son características importantes de RF (Ziegler y König, 2014).

Este algoritmo adopta un conjunto de árboles de decisión y determina las clases categóricas mediante un algoritmo de voto mayoritario. Por lo tanto, es necesario considerar seriamente el sobreajuste (en inglés *overfitting*) para probar el rendimiento de RF. Normalmente, se producirá un ajuste excesivo cuando el aprendizaje se lleve a cabo durante demasiado tiempo o cuando los ejemplos de formación sean atípicos; el algoritmo puede estar limitado a características aleatorias muy específicas de los datos de entrenamiento que no tienen relación causal con la función objetivo. De todas formas, con RF, es posible evitar generalmente el sobreajuste (Breiman y cols., 1984).

El riesgo de *overfitting* se reduce sustancialmente debido a dos aspectos del algoritmo. La primera es que las muestras son remuestreadas (*bootstrapped* en inglés) para generar los árboles individuales, y *bagging* reduce el riesgo de *overfitting*, y el segundo es el componente aún más importante de que se selecciona aleatoriamente un pequeño conjunto de características por árbol (Schwarz, König, y Ziegler, 2010). Para determinar el error de predicción, Breiman primero hizo crecer el bosque aleatorio en el 90 % de los datos que se seleccionaron al azar, y

el otro 10 % de los datos se apartó del bosque aleatorio para obtener un error de conjunto de prueba (Breiman, 2001).

5.8.2. K-Nearest-Neighbor-KNN

Es un algoritmo que permite implementar una clasificación no-paramétrica que usa conjunto de datos de entrenamiento para clasificar nuevos datos o datos de prueba, con el fin de aplicar el criterio del vecino más cercano. La forma de la que funciona este algoritmo consiste en encontrar k muestras de la base de datos de entrenamiento más cercanos a la muestra de prueba, El etiquetado se basa en las clases o etiquetas que predominen en el vecindario (X. Wu y cols., 2007).

Para clasificar un objeto sin etiquetar se realiza según G. Guo, Wang, Bell, y Bi (2004) :

1. Calcular la distancia: Dado el objeto de prueba, calcule la distancia desde cada objeto en el conjunto de entrenamiento.

2. Encontrar k vecinos: La delimitación tanto de los k objetos de entrenamiento más cercanos como del objeto de prueba de los vecinos.

3. Clasificación: Una vez se identifican los k vecinos más cercanos, y las etiquetas de clase de estos vecinos más cercanos se utilizan para determinar la etiqueta de clase del objeto.

De esta forma, el algoritmo de KNN queda definido de la Ecuación (33):

$$y' = \underset{v}{argmax} \sum_{(xi,yi)\in D} W_i \times I(v = y_i) \tag{33}$$

Donde v es la etiqueta de la clase, yi es la etiqueta perteneciente a la clase del i-ésimo vecino más cercano, $W_i = \frac{1}{d(x',x_i)^2}$, es un factor de ponderación que está en función del cuadrado inverso de su distancia y $I()$ es una función que toma el valor de 1 cuando el argumento es verdadero, pero si es 0 cuando no lo es.

Profundizando en el algoritmo, se debe mencionar que este método no paramétrico muy simple se ha utilizado en tareas de clasificación (Halkidi y Vazirgiannis, 2002), (Cover y Hart, 1967). No requiere un paso de entrenamiento específico, por lo tanto, para clasificar una nueva muestra de entrada, se clasifican los vecinos más cercanos del conjunto de datos de entrenamiento y luego, la clase más común se asigna a la nueva muestra. Este método básicamente se enfoca en las etiquetas de los vecinos más cercanos. Aplicando esto al trabajo

realizado, este algoritmo utiliza datos de entrenamiento como fundamento al clasificar de nuevos datos o muestra que pertenecen al conjunto de prueba, aplicando el criterio del vecino que más cerca se encuentre. Este enfoque se basa en encontrar cantidad de k muestras del conjunto de entrenamiento más cercanas a nueva muestra de prueba (Sánchez Loja, 2018).

Funciona usando dos instancias con atributos p, $x_i \{= x_{i1}, x_{i2}, ..., x_{i,p}\}$ y $x_j \{= x_{j1}, x_{j2}, ..., x_{j,p}\}$ de el conjunto de datos de entrenamiento, donde $i = 1, 2, ..., n$, $j = 1, 2, ..., n$, y n es el número total de muestras. La distancia entre ellos se define mediante una métrica (xi, xj); esta métrica puede ser euclidiana, Manhattan, coseno, chi-cuadrado, entre otras medidas de distancia (Dasarathy, 1991). Básicamente, para etiquetar una nueva clase, el algoritmo KKN encuentra vecinos del conjunto de datos de entrenamiento, con las distancias más cortas según la métrica elegida; luego, KNN asigna la clase dominante entre los vecinos más cercanos (Sanchez y cols., 2019).

En cuanto a los clasificadores basados en este algoritmo, se puede relacionar con el principio más simple para describir el aprendizaje basado en instancias, el cual es que las instancias similares tienen etiquetas de clases similares. Es por esta razón que el enfoque natural para aprovechar este principio general es utilizar clasificadores de vecino más cercano (*nearest neightbor* en inglés). Para una instancia de prueba determinada, se determinan las k muestras de entrenamiento más cercanas (Samanthula, Elmehdwi, y Jiang, 2014). La etiqueta dominante entre estas k muestras de entrenamiento se informa como la clase relevante. En algunas variaciones del modelo, se utiliza un esquema de ponderación de distancia inversa, para tener en cuenta la importancia variable de las k instancias de entrenamiento que están más cerca de la instancia de prueba. Un ejemplo de una función de peso inversa de la distancia δ es $f(\delta) = e^{-\delta^2/t^2}$, donde t es un parámetro definido por el usuario y δ es la distancia del punto de entrenamiento a la instancia de prueba. Este peso se utiliza como voto y la clase con el mayor número de votos se informa como la etiqueta correspondiente. Si se desea, se puede construir un índice de vecino más cercano al principio, para permitir una recuperación más eficiente de instancias (Aggarwal, 2015).

El mayor desafío con el uso del clasificador del vecino más cercano es la elección del parámetro k. En general, un valor muy pequeño de m no conducirá a resultados de clasificación robustos debido a variaciones ruidosas dentro de los datos. Por otro lado, los valores grandes de m perderán sensibilidad a la localidad de datos subyacente. En la práctica, el valor apropiado de m se elige de forma heurística. Un enfoque común es probar diferentes

valores de k para precisión sobre los datos de entrenamiento. Por último, se debe mencionar que estos clasificadores se basan en la memoria (*memory-based* en inglés) y por lo general, no requieren ningún modelo para ajustarse, aunque es posible implementarlo en casos donde el rendimiento del modelo no sea el adecuado. Reiterando, dado un punto de consulta x_0, encontramos los k puntos de entrenamiento $x_{(r)}, r = 1, ..., k$ más cercanos en distancia a x_0, y luego clasificamos usando el voto mayoritario entre los k vecinos (Franklin, 2005).

En cuanto a la clasificación de este algoritmo, implica dividir las muestras en categorías de entrenamiento y prueba. Sea $x_i|$ una muestra de entrenamiento y $x|$ una muestra de prueba, y sea $\omega|$ la verdadera clase de una muestra de entrenamiento y $\hat{\omega}|$ la clase predicha para una muestra de prueba ($\omega, \hat{\omega} = 1, 2, ...\Omega$), donde Ω es número total de clases. Ahora, durante el proceso de entrenamiento, se usa solo la clase verdadera $\omega|$ de cada muestra de entrenamiento para entrenar al clasificador, mientras que durante las pruebas se predice la clase $\hat{\omega}|$ de cada muestra de prueba. De todos modos, KNN no garantiza nada, aunque sea un método de clasificación "supervisado.en el sentido de que utiliza las etiquetas de clase de los datos de entrenamiento (Peterson, 2009).

Para los k vecinos más cercanos, la clase predicha de la muestra de prueba x se establece igual a la clase verdadera más frecuente entre las k muestras de entrenamiento más cercanas. Esto forma la regla de decisión $D : x \to \hat{\omega}$. La matriz de confusión utilizada para tabular las predicciones de la clase de muestra de prueba durante la prueba se indica como C y tiene dimensiones $\Omega \times \Omega$. Durante la prueba, si la clase predicha de la muestra de prueba x es correcta (es decir, $\omega = \hat{\omega}|$), entonces el elemento diagonal de la matriz de confusión se incrementa en 1. Sin embargo, si la clase predicha es incorrecta (es decir, $\omega \neq \hat{\omega}|$), entonces el elemento fuera de la diagonal $c\omega\omega|$ se incrementa en 1. Una vez que se han clasificado todas las muestras de prueba, la precisión de la clasificación se basa en la relación entre el número de muestras correctamente clasificadas y el número total de muestras clasificadas, dado en la forma según Peterson (2009) y en la Ecuación 34:

$$Acc = \left| \frac{\sum_{\omega}^{\Omega} c\omega\omega}{n_{total}} \right| \tag{34}$$

donde $c\omega\omega$ es un elemento diagonal de C y n_{total} es el número total de muestras clasificadas. Para propósitos de entrenamiento, se usó la validación ajustada. El rendimiento de la mayoría de los clasificadores se evalúa típicamente a través de la validación cruzada (cross-validation, en inglés), que implica la determinación de la precisión de la clasificación para múltiples

particiones de las muestras de entrada utilizadas en el entrenamiento (Peterson, 2009).

5.8.3. Ajuste de modelos y clasificación de fallos con los algoritmos KNN y Random Forest

Como se ha mencionado anteriormente, con los atributos ya seleccionados es posible proceder a obtener el mejor clasificador basado en KNN y RF, con los datos de entrenamiento. Después del entrenamiento de los algoritmos es posible aplicar los datos de prueba y poder clasificar cada muestra en su determinado nivel de severidad de fallo del diente roto. Puesto que son dos algoritmos diferentes, se menciona brevemente lo que se puede realizar en caso de que los resultados de la clasificación no tengan una alta precisión.

- Para **ajustar un clasificador KNN** se debe tomar un conjunto de datos como entrada y luego sacar un clasificador, que se elige de un espacio de posibles clasificadores. Los parámetros usados para KNN son identificados por los propios datos de entrenamiento (Aggarwal, 2015). Entonces, para ajustar un clasificador KNN simplemente requiere almacenar el conjunto de entrenamiento. El número de vecinos k y la métrica de distancia son hiperparámetros de clasificadores KNN. Por lo general, el rendimiento se puede mejorar eligiéndolos de forma que se adapten al problema. Pero, las configuraciones óptimas generalmente no se conocen de antemano, y deben establecerse durante el procedimiento de entrenamiento (Peterson, 2009). También es posible aplicar validación cruzada a este algoritmo al generar varios modelos mediante el remuestreo de datos perteneciente a conjunto de datos limitados.

- Para **ajustar un bosque aleatorio**, se reduce a seleccionar hiperparámetros. Los hiperparámetros de un RF son los hiperparámetros del estimador base subyacente (de nuevo, típicamente un árbol de decisión) que podrían ser numerosos. La cantidad de árboles que se van a poner en el bosque llega a ser un hiperparámetro importante dentro del ajuste. Los hiperpárametros son parámetros establecidos por el científico de datos antes del entrenamiento. A medida que aumenta el número de árboles, la varianza disminuye. En Random Forest hay formas de acelerar la selección de hiperparámetros y una de ellas es mediante la validación cruzada OBB (en inglés *out-of-bag cross validation*) u OOB cross validation. El OOB se utiliza para predecir y evaluar los resultados por modelo entrenado. Al comparar los resultados y las observaciones, se puede acumular el error de clasificación o el error de prueba(X. Guo y Hao, 2021). Para la **validación cruzada del OOB**, se debe mencionar que cuando se construye un árbol aleatorio, el procedimiento de arranque descrito anteriormente significa que solo una fracción de los

datos de entrenamiento se incluye en los datos utilizados para ajustarse a ese árbol en particular(Mitchell, 2011).

6. Marco metodológico

Esta sección ha sido dividida en tres partes que consisten en: Metodología de la investigación, Método del proceso y Método estadístico. Cada una de estas partes se explican a continuación.

6.1. Metodología de la Investigación

La investigación realizada es de tipo básica y cuantitativa, puesto que se busca la obtención de nuevo conocimiento y se emplearon datos numéricos obtenidos del grupo de investigación GIDTEC. Este proyecto parte de actividades desarrolladas previamente, como el análisis exploratorio de la señal del par eléctrico como trabajo de investigación de pasantía realizado por la autora Mejía (2022). En el trabajo de Ortega Lucero (2021), se midió las señales de corriente y voltaje y mediante un modelo matemático se determinó el par eléctrico. Lo que se busca con esta señal es obtener los atributos de Poincaré y después mediante métodos de *Ranking* de atributos como CDbw, Random Forest y ReliefF para definir los atributos más relevantes. Es necesario mencionar que la base de datos fue dividida en: un dataset de entrenamiento y otro dataset de prueba. Este trabajo se enfoca en usar los mejores atributos de Poincaré y usarlos en algoritmos de aprendizaje automático llamados KNN y RF para comenzar una fase de entrenamiento. Posteriormente, con los algoritmos ya entrenados va a ser posible clasificar los datos de prueba y determinar su precisión, exactitud, etc. En cuanto a los alcances, se aplica lo exploratorio y descriptivo, porque se utilizaron los algoritmos de aprendizaje automático KNN y RF para obtener un modelo computacional para clasificar el nivel de severidad de rotura de un diente mediante el par eléctrico. Esta señal ha sido poco estudiada para detectar fallos y el uso los atributos de Poincaré como indicadores de los modelos tampoco es muy común, lo cual permite generar nuevos resultados y conocimientos a analizar y aplicar para la detección de fallo de nivel diente roto.

6.2. Metodología del proceso

ENFOQUE:

INVESTIGACIÓN CUANTITATIVA:

Cuantitativa: Análisis de los resultados de precisión obtenida por la clasificación de RF y KNN.

TIPO:

BÁSICA:

En trabajo permite la generación de nuevos conocimientos en campo del mantenimiento basado en la condición y aplicación de algoritmos de aprendizaje automático.

ALCANCES:

EXPLORATORIO

Identificar los atributos de Poincaré adecuados para el el entrenamiento de algoritmos de aprendizaje automático KNN y RF.

DESCRIPTIVO

Especificación de los atributos aplicados al modelo y determinar la precisión de los mismos (KNN y RF).

En esta sección se explica el procesamiento de datos y parte de los resultados obtenidos del *Ranking* de atributos con ReliefF y CDbw obtenidos del análisis exploratorio de los atributos de Poincaré de la señal de par eléctrico en Mejía (2022). Por otro lado, también se explica la metodología para el entrenamiento tanto para KNN y RF junto con los diferentes parámetros seleccionados y su influencia en los resultados finales. Por último, también se explica el *Ranking* de atributos obtenidos con RF y a importancia obtenida para cada atributo.

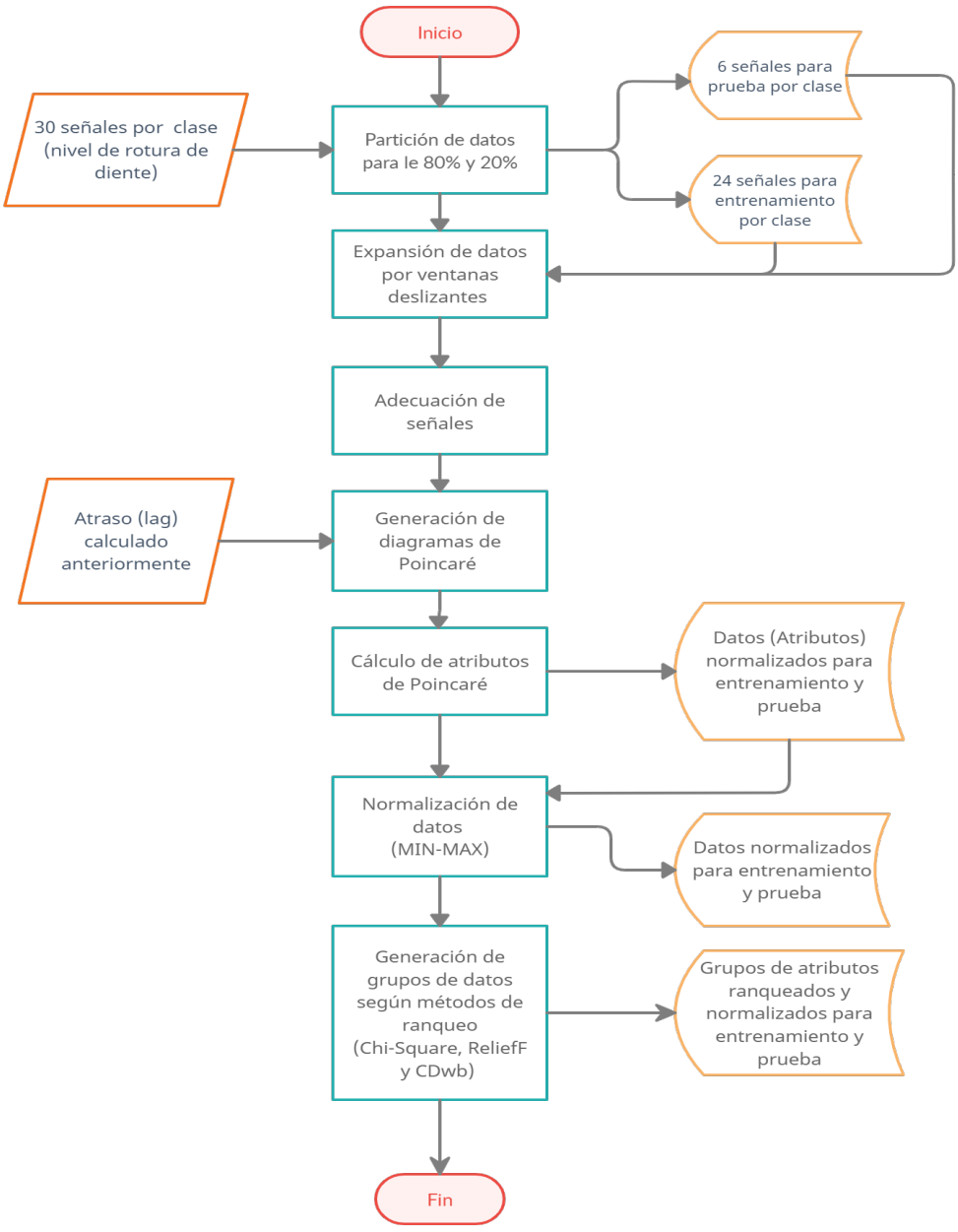
6.2.1. Metodología para procesamiento de datos

En la Figura 16 se observa un diagrama de flujo que indica el procedimiento seguido para procesar la señal del par eléctrico. Debido a que todo esto fue desarrollado y descrito con mayor detalle en el trabajo de Mejía (2022), lo presentado a continuación es un resumen de dicho trabajo para tener una idea general del proceso realizado con la señal de par eléctrico antes de usarla en algoritmos de entrenamiento.

De este modo, antes de comenzar de con el entrenamiento de los algoritmos KNN y RF

Figura 16

Diagrama de flujo para procesamiento de datos.



Nota: *En este diagrama se observa los procesos usados para el procesamiento de la señal de par electrico (Mejía, 2022).*

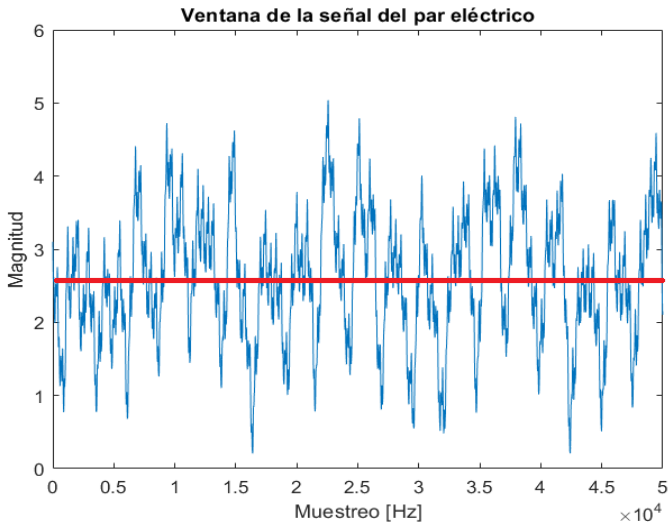
para la generación de diferentes modelos computacionales, es necesario hacer un tratamiento de datos que consiste en primer lugar en una **partición de datos entre 80 % y 20 %**; sin embargo, un aspecto a tener en cuenta es que los datos con los que se está trabajando fueron reducidos anteriormente debido al ruido existente en cada una de las señales durante los primeros 0,05 segundos. La duración de las señales fue reducida de 10 segundos a 9,95 segundos con el fin de eliminar este ruido inicial. En otras palabras, se redujeron, 25000 muestras de 500000 muestras en total de todas las señales de la base de datos original, generando así la base de datos con la cual se está realizando el entrenamiento y prueba del algoritmo. Una vez considerado este aspecto, el 80 % de los datos fueron usados para el entrenamiento del algoritmo y el 20 % fueron usados para probar los modelos entrenados por el algoritmo. En total se trabajó con 30 señales por nivel de fallo y en este caso, hay 9 niveles, por lo tanto, se dispuso 270 señales en total. Al realizar la partición de datos del 80 % y 20 % se obtienen 24 señales para entrenamiento y 6 para la prueba por cada nivel de fallos, lo cual significa 216 señales para entrenamiento y 54 para prueba.

Después se realizó una **expansión de datos por ventanas** deslizantes (*data augmetation* en inglés) tanto para los datos de entrenamiento y prueba porque los datos disponibles no eran los suficientes para generar resultados estadísticamente significativos para su posterior análisis. Durante este proceso de expansión de datos, se aplicaron ventanas deslizantes sobre cada señal de par de eléctrico para generar nuevas señales de una duración mucho más corta que la señal original (subseñales) y con un cierto solapamiento entre las ventanas aplicadas. En este caso, se determinó un intervalo de solapamiento entre ventanas de 0,1 segundos (equivalente a 5000 muestras) y una duración para cada ventana de 1 s (equivalente a 50000 muestras). Aplicando este procedimiento se generaron 90 subseñales por cada señal original, por lo tanto, esto permitió obtener 19440 datos de entrenamiento y 4860 datos de prueba. En la Figura 17 se observa el ejemplo de una señal ventaneada.

En la Figura 17 se puede observar una señal del par eléctrico y con un offset aproximado de 2 unidades en su magnitud, por lo que es necesario realizar un ajuste o **adecuación de todas las señales** de la base datos para encontrar un retraso o *lag* (τ) apropiado. En general, una gráfica de Poincaré se obtiene seleccionando correctamente un *lag* (τ), que es conocido cuando el valor para el cual la autocorrelación de $x(t+\tau)$ es cero o cerca de cero. La forma de encontrar el *lag*, se realizó mediante del cálculo la media para todas las señales disponibles, ya sean de entrenamiento o de prueba, y se restó este valor a la respectiva señal, con el objetivo de cambiar el offset de la señal a 0 como se observa en la Figura 18 porque resulta

Figura 17

Ventana de una señal de par eléctrico.



Nota: *Ejemplo de una subseñal de par eléctrico con un offset (Mejía, 2022).*

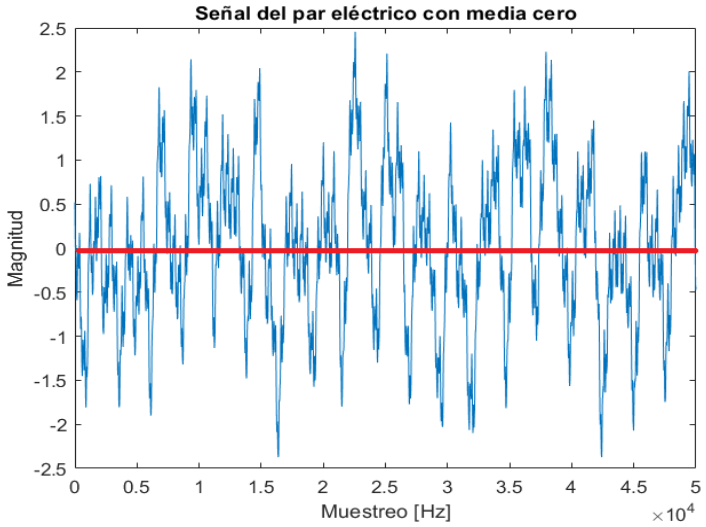
inconveniente es valor de offset al calcular la autocorrelación. Con las señales ajustadas y conociendo el atraso apropiado es posible seguir con el proceso.

Para la **generación de los diagramas de Poincaré**, se aplicó un *lag* de 726, este valor fue obtenido anteriormente en el trabajo de pasantía de la autora Mejía (2022). La forma de los diagramas de Poincaré (por lo general una nube de puntos) varía según el nivel de severidad del fallo de rotura de diente y a partir de estos diagramas es posible calcular determinadas características de los mismos como la asimetría, etc. En otras palabras, es posible realizar el **cálculo de los 11 atributos de Poincaré** los cuales fueron explicados en el marco teórico, específicamente en la sección 5.6.

Se debe resaltar nuevamente que los atributos de Poincaré fueron calculados tanto para los datos de entrenamiento y prueba y que todos estos datos tuvieron que ser normalizados. Para **la normalización de los datos** se determinaron los valores máximos y mínimos (valores entre 0-1) para cada uno de los atributos de todos los niveles de severidad. La técnica de normalización empleada fue el min-max que generalmente fue utilizada para procesar imágenes; esta técnica tiene propiedades para re-escalar magnitudes de cada atributo

Figura 18

Ventana de una señal de par eléctrico con media cero.



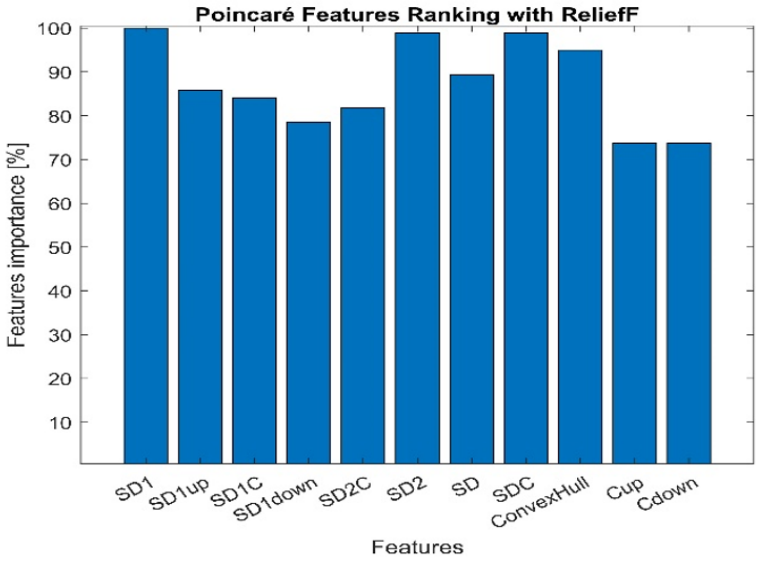
Nota: *Ejemplos de subseñal de par eléctrico con media 0 (Mejía, 2022) .*

usando un rango (0 – 1), con el fin de trabajar valores positivos. Es necesario establecer el valor mínimo y máximo presente de cada atributo para el conjunto de entrenamiento y de prueba.

En el trabajo de pasantía de análisis exploratorio de datos de la autora usado para esta proyecto técnico, se usó la base de datos del par eléctrico y se clasificó mediante diferentes métodos los atributos de Poincaré más importantes. Los métodos usados fueron Chi Cuadrado, CDbw y ReliefF, pero en este trabajo se implementó Random Forest y se descartó Chi Cuadrado. A partir de esto se **generaron diferentes grupos de atributos según el método de clasificación de orden** de importancia. Estos grupos generaron desde 3 atributos hasta de 10 atributos dentro del conjunto ordenado del más importante, al menos relevante, por lo tanto, hay 8 grupos de atributos por método de *Ranking* y se incluye, un último grupo, pero este consiste de todos los atributos de Poincaré. De este modo, al iniciar este trabajo, hubo 16 grupos de atributos usados como parámetro a modificar dentro de los modelos generados, pero este número aumenta al implementar RF dando 24 grupos en total. Los resultados del *Ranking* según CDbw y ReliefF se presentan en la Figura 19 y en la Tabla 5.

Figura 19

Ranking con ReliefF de atributos de Poincaré.



Feature	Importance
SD1	100 %
SD1up	85,806 %
SD1C	84,029 %
SD1down	78,560 %
SD2C	81,768 %
SD2	98,892 %
SD	89,273 %
SDC	98,897 %
Convex Hull	94,957 %
Cup	73,716 %
Cdown	73,716 %

Nota: *Ranking con ReliefF de atributos de Poincaré con sus respectivos procentajes de importancia del lado derecho (Mejía, 2022).*

Tabla 5

Atributos seleccionados por CDbw según su cantidad.

Ranking de atributos por CDbw	
Grupo	Atributos
Grupo 1	[5,11,4]
Grupo 2	[5,6,11,4]
Grupo 3	[5,10,2,8,4]
Grupo 4	[3,10,2,7,8,4]
Grupo 5	[3,1,10,2,7,8,4]
Grupo 6	[3,1,6,10,2,7,8,4]
Grupo 7	[5,1,6,11,10,9,2, 7,8]
Grupo 8	[3,1,6,11,10,9,2,7,8,4]

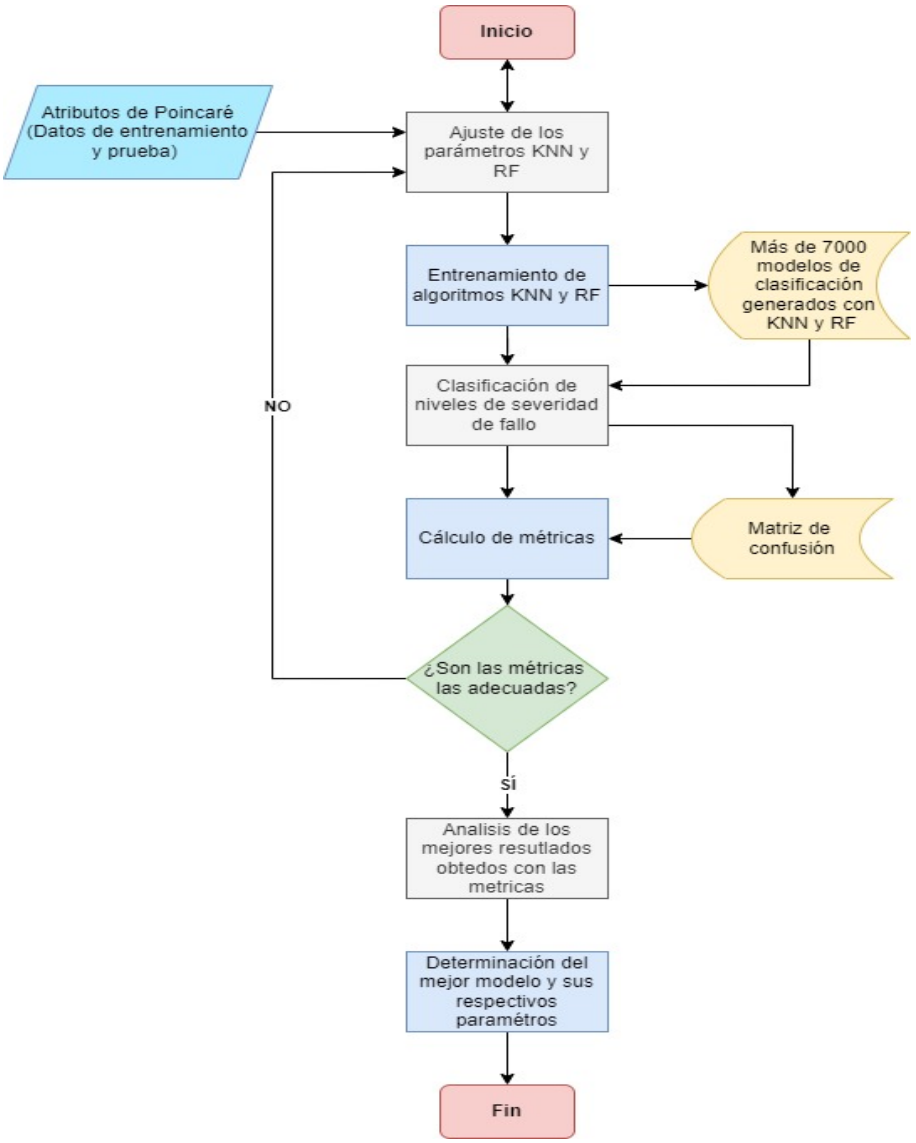
Nota: *Ranking con CDbw de atributos de Poincaré dependiendo de su importancia y cantidad (Mejía, 2022).*

6.2.2. Metodología general para entrenamiento de modelos de aprendizaje automático

En la Figura 20 se presenta el proceso para el desarrollo de este trabajo de titulación en forma de diagrama de flujo y a lo largo de esta sección se describen los procesos mostrados para la configuración y generación de modelos de clasificación y medir su rendimiento.

Figura 20

Proceso general de investigación.



Nota: *En este diagrama se observa los procesos usados para la generacion de modelos entrenados usando los algoritmos KNN y RF.*

En primer lugar, se realizó un **ajuste de parámetros de RF y KNN**, donde el objetivo, por lo general, es obtener el mejor rendimiento para los modelos generados. En el caso del algoritmo KNN, tal como se ha explicado en el marco teórico, el ajuste significa tomar un conjunto de datos como entrada y luego generar un clasificador. Los parámetros usados se eligen típicamente resolviendo un problema de optimización o algún otro procedimiento numérico. Para este trabajo de titulación, se seleccionaron diferentes parámetros de entrenamiento, como: el número de k vecinos, que tenían un rango de valores de 46 a 56 (11 números en total). La razón de haber seleccionado un valor desde 46 vecinos es por la raíz cuadrada de la cantidad de datos disponibles para el entrenamiento, es decir, 19440 subseñales. Otro de los parámetros es la distancia con la cual se entrenó a los algoritmos. En este caso, se escogió la distancia euclidiana, coseno y Mahalanobis. Por otro lado, para obtener resultados más confiables se realizó la validación cruzada de los datos de entrenamiento, generando de esta forma 10 modelos diferentes. La validación cruzada es un método estadístico para evaluar y comparar algoritmos de aprendizaje dividiendo los datos en dos segmentos: uno usado para entrenar un modelo y el otro usado para validar el modelo. Otro parámetro más que se debe mencionar es la cantidad de atributos de Poincaré usados para el entrenamiento de KNN. Como se menciona en la sección 6.2.1, se generaron 8 diferentes grupos de atributos por método de *Ranking,* ya sea CDbw, ReliefF o Random Forest. Estos grupos constan desde 3 hasta 10 atributos, lo cual da significa que en total se tienen 24 grupos en total como parámetro a modificar al entrenar el algoritmo KNN.

Por otro lado, para ajustar los parámetros de RF, es fundamental establecer la cantidad de árboles disponibles en el bosque, número de iteraciones, cantidad de padres e hijos, entre otros. También se realiza la validación cruzada pero específicamente se realiza una **validación cruzada del OOB**, la cual también ayuda a selección de parámetros de forma más rápida. Se menciona nuevamente que al construir un árbol aleatorio, el procedimiento de arranque significa que solo una fracción de los datos de entrenamiento se incluye en los datos utilizados para ajustarse a ese árbol en particular y el resto de datos debe usado para realizar validación. En este caso, RF fue entrenado con parámetros como: el número de árboles (se seleccionó un rango desde 5 árboles que aumenta cada 5 árboles hasta llegar a 65 árboles), el número mínimo de hojas (en este caso es de 1), activación del *Bootstraping*, el número mínimo de padres (2 padres), entre otros para la programación en Matlab. Se debe mencionar como parámetro a modificar, también se encuentran los diferentes grupos de atributos con los que se entrenan los bosques aleatorios. Los grupos de atributos fueron obtenidos por el *Ranking* obtenido por ReliefF, CDbw y por RF.

Después de ajustar los parámetros, se realiza el **entrenamiento de algoritmos KNN y RF** para generar modelos computacionales de aprendizaje automático. Anteriormente, se realizó un análisis exploratorio de datos donde se había separado los atributos de Poincaré de la base de datos del par eléctrico, en datos de prueba y datos de entrenamiento, este último mencionado representa la entrada del proceso de entrenamiento, el cual consiste en proveer a un algoritmo de aprendizaje automático datos para identificar y reconocer valores relevantes de los patrones involucrados en el proceso. En el caso de este trabajo, se usaron algoritmos de entrenamiento supervisados como KNN y RF, lo que significa que se requiere de datos etiquetados para que realizar el entrenamiento. En rasgos generales, se puede decir que para el entrenamiento de KNN se utiliza un conjunto de datos de entrenamiento como base para realizar la clasificación de nuevas muestras, pertenecientes al conjunto de prueba, empleando el criterio del vecino más cercano. Consta de tres actividades principales que son: Calcular la distancia del objeto dado, encontrar k vecinos más cercanos y por último realizar la clasificación de datos. Para RF o un bosque aleatorio, se generan varios árboles de decisión hasta obtener un bosque aleatorio. Al momento del entrenamiento, cada árbol emite un voto unitario y la clase más popular permite clasificar los datos. Se puede encontrar más detalles sobre el entrenamiento en las secciones 5.8.1 y 5.8.2.

Un aspecto que es necesario mencionar es que por cada número de k vecinos para KNN y cantidad de árboles dentro del bosque aleatorio, se generaron sus respectivos modelos. La cantidad de modelos resultantes dependen de los parámetros ajustados. Para KNN se obtuvieron 10 modelos entrenados por valor de k vecinos debido a la validación cruzada y como este valor de k tiene un rango de 46 a 56 (11 números en total), se generaron como máximo 110 modelos entrenados usando un determinado grupo de atributos (cantidad de atributos) dependiendo del método de *Ranking*. Puesto que se usaron 8 diferentes grupos de atributos por método de *Ranking*, el máximo de modelos obtenidos por distancia sería de $(110 \times 8 = 880)$ y debido a que los métodos de *Ranking* son 3 (CDbw, Random Forest y ReliefF) se obtiene que $880 \times 3 = 2640$ es la cantidad de máxima de modelos obtenidos según distancia. Ahora, incluyendo las tres diferentes distancias de KNN (Euclidiana, Coseno y Mahalanobis) la cantidad máxima final de modelos obtenidos es de $2640 \times 3 = 7920$ modelos. La cantidad de modelos es una estimación debido a que aún existía empate de la suma de probabilidad posterior de las subseñales, por lo tanto, no todos los modelos generados eran válidos. También ocurrió que al entrenar al algoritmo con 10 atributos del *Ranking* de CDbw y de RelieFf usando la distancia Mahalanobis no se generó ningún modelo porque la matriz de covarianza requerida para esta distancia no era positiva definida. De esta forma, es posible

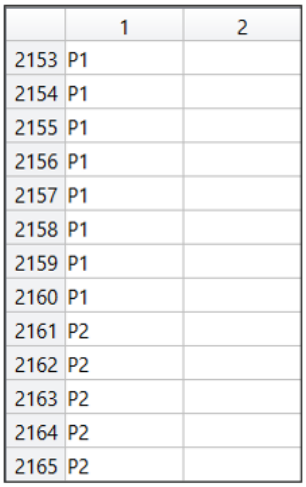
decir que al menos 7000 de los 7920 modelos que debían generarse en caso de no existir ningún inconveniente, son válidos y permiten clasificar los diferentes niveles de severidad de diente roto tanto para datos de entrenamiento y prueba.

En cuanto a RF, el número de árboles para el bosque tenía un rango de 5 a 65 (donde cada valor se saltaba cada 5 números, ejemplo: 5,10,15,etc.), por lo tanto, para este algoritmo el rango del uno de sus parámetros consta de 11 números, dando lugar a 11 modelos por grupo de atributos según el método de *Ranking*. De este modo, al existir 8 grupos y 3 diferentes métodos de *Ranking* se tiene que $11 \times 8 \times 3 = 264$ y a este valor se le suma 11 porque se incluye el grupo de todos los atributos de Poincaré, en este caso no importa el método de *Ranking* usado porque se trata de la cantidad máxima de atributos disponibles. El total de modelos generados por RF es de 275 modelos. Se concluye que para los modelos generados a partir de KNN y RF se obtuvieron más de 7000 matrices de confusión porque cada matriz es propia de cada modelo y de todos los modelos generados debe determinarse cuál obtuvo las mejores métricas y cuáles fueron los parámetros ajustados dependiendo del algoritmo implementado.

A partir de los diferentes modelos entrenados y de disponer tanto de datos de entrenamiento y prueba, en un principio se entrenó y se clasificaron solamente los datos de entrenamiento con el fin de conocer el rendimiento de los modelos generados. Una vez determinado si los modelos está en condiciones de clasificar nuevos datos, se proporcionaron los datos de prueba. Todos estos procesos se enfocan en la **clasificación de niveles de severidad de fallo de diente roto** y una vez explicada la razón de no solo se clasifican los datos de prueba, pero también los de entrenamiento, se menciona que al usar los comandos propios de Matlab para los resultados de predicción de cada modelo generan dos matrices de clasificación: Una matriz de etiquetas, la cual solo posee una columna mostrando a qué nivel de fallo pertenecen las muestras o cada subseñal. Las etiquetas van desde P1 hasta P9, puesto que son los niveles de severidad de diente roto como se observa en la Figura 21. También se obtiene una matriz de probabilidad posterior. La probabilidad posterior hace referencia a la probabilidad que tienen los datos de ser clasificados con determinada etiqueta. Matlab genera una matriz que contiene 9 columnas debido a los 9 niveles de severidad de fallo, por lo tanto, se obtiene una fila con la probabilidad de cada nivel de severidad de fallo y el valor más alto de la fila determina la etiqueta con la que se clasifica una subseñal. Un ejemplo de esto se puede ver la Figura 22.

Figura 21

Matriz de etiquetas una vez realizada la clasificación.



	1	2
2153	P1	
2154	P1	
2155	P1	
2156	P1	
2157	P1	
2158	P1	
2159	P1	
2160	P1	
2161	P2	
2162	P2	
2163	P2	
2164	P2	
2165	P2	

Nota: *Ejemplo de matriz de etiquetas generadas con los codigos de Matlab.*

Figura 22

Matriz de etiquetas una vez realizada la clasificación.

Ventana/ Nivel severidad	P1	P2	P3	P4	P5	P6	P7	P8	P9
1	1	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
2	1	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
3	1	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
4	1	0,000	0,000	0,000	0,000	0,000	0,000	0,000	0,000
5	0,664	0,137	0,000	0,000	0,000	0,199	0,000	0,000	0,000
6	0,99	0,000	0,000	0,000	0,000	0,010	0,000	0,000	0,000
7	0,750	0,025	0,000	0,000	0,000	0,225	0,000	0,000	0,000
8	0,237	0,681	0,000	0,000	0,000	0,082	0,000	0,000	0,000
9	0,829	0,029	0,000	0,000	0,000	0,142	0,000	0,000	0,000
10	0,387	0,001	0,000	0,000	0,000	0,613	0,000	0,000	0,000
11	0,396	0,000	0,000	0,000	0,000	0,604	0,000	0,000	0,000
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
90	1	0	0	0	0	0	0	0	0

Nota: *Ejemplo de matriz de porbabilidad de matriz posterior.*

Como siguiente paso, es posible poner a prueba los modelos entrenados con sus respectivos datos para determinar un porcentaje de precisión, exactitud, etc. en su clasificación mediante el **cálculo de la matriz de confusión**. Esta matriz es una herramienta que permite la visualización del desempeño de la clasificación dentro del aprendizaje supervisado y sirve para mostrar de forma explícita cuándo una clase es confundida con otra. Esta matriz consta de varias métricas tales como la precisión, indicadores como F1-score, falsos positivos, entre otros. En este trabajo se definieron las métricas de exactitud, precisión, recall y F1-score calculadas a partir de la matriz de confusión.

Con relación al cálculo de la matriz de confusión, en un principio, se decidió aplicar directamente los comandos de Matalb para generarla mediante la matriz de etiquetas anteriormente mencionada, pero la clasificación de datos no se superaba el 54 % de precisión y exactitud. Debido a que los resultados no eran los deseados, se decidió contar el número de etiquetas por cada 90 ventanas o subseñales pertenecientes a una original, debido a que las ventanas podían ser fácilmente confundidas entre clases. El método usado se basaba en que la señal con mayor número de subseñales etiquetadas desde P1 hasta P9 determinada la etiqueta de la señal completa (Léase sección 6.2.1 para mejor compresión de ventanas). El principal problema de realizar el conteo de etiquetas correspondiente a las ventanas o subseñales es que existía empate en determinados modelos obtenidos, ya sea por entrenamiento de RF o KNN. De este modo, para evitar el empate se optó por la suma de la probabilidad posterior por cada 90 subseñales. La probabilidad posterior consiste en una probabilidad condicional dentro de la estadística bayesiana y hace referencia a la probabilidad de determinado evento que se actualiza sobre una variable aleatoria después de tomar en consideración algunos datos (Winn, Bishop, y Diethe, 2015). Esta probabilidad está definida por la ecuación (35):

$$P(A \mid B) = \frac{P(B \mid A) \times P(A)}{(P \mid B)} \tag{35}$$

donde:

$P(A \mid B)$ es la probabilidad posterior.

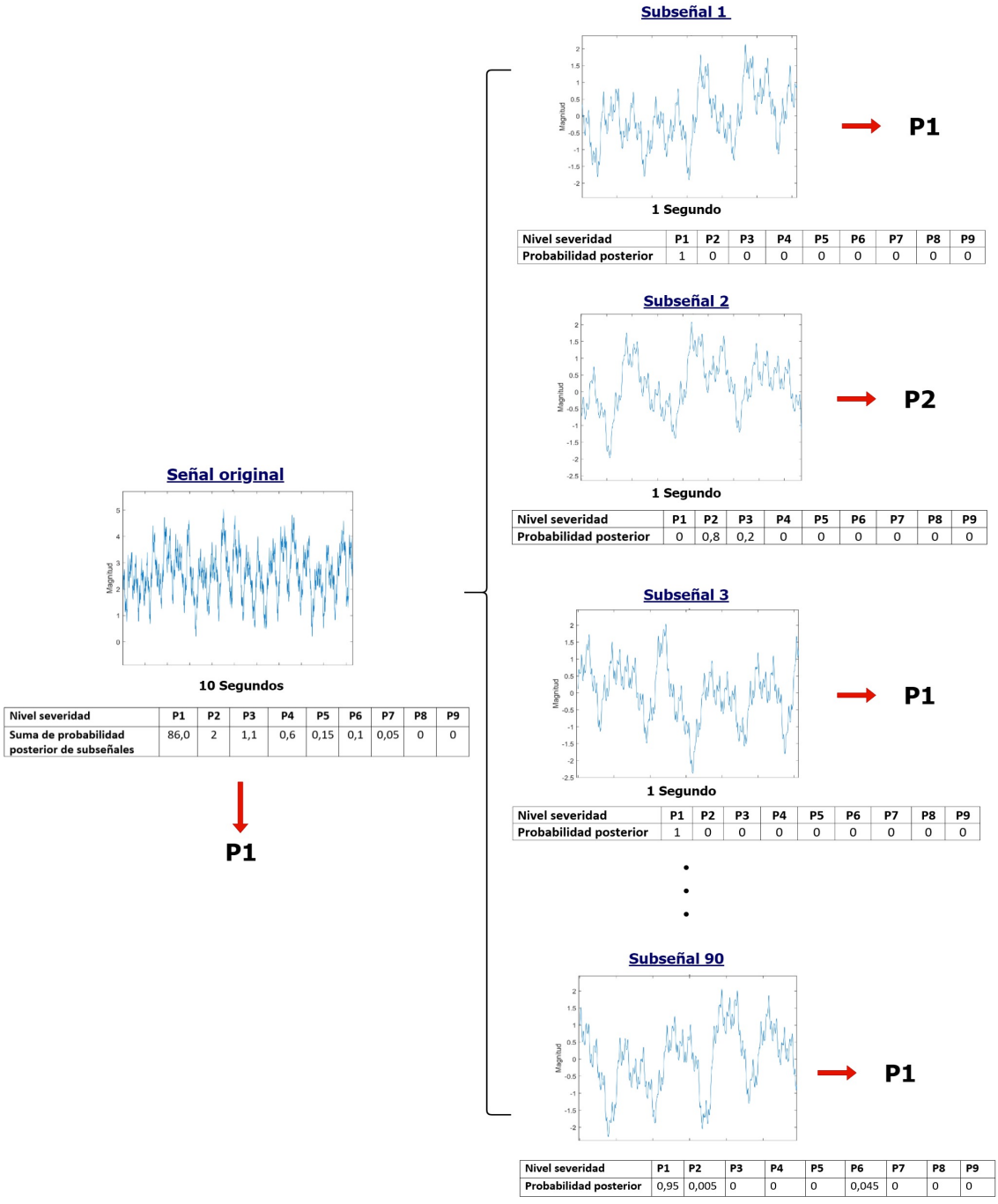
A, B son los eventos.

$P(B \mid A)$ es probabilidad de que ocurra B dado que A es verdadera.

$P(A)$ y $P(B)$ son las probabilidades de que ocurra A y B independientemente una de la otra.

Figura 23

Proceso para etiquetar señales originales.



Nota: *A partir de la suma de probabilidad porsterio para cada nivel de fallo de 90 subseñales.*

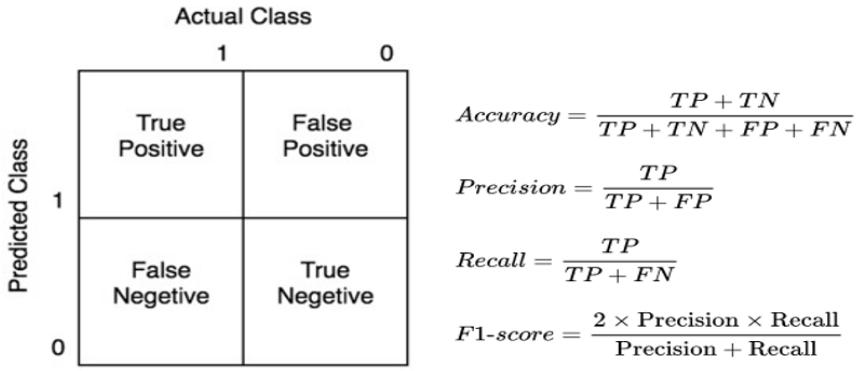
Después de definir la probabilidad posterior, se debe mencionar que cada subseñal tiene determina probabilidad de ser etiquetada en cualquier nivel de severidad de fallo de diente roto (P1-P9). La mayor probabilidad obtenida respecto al determinado nivel de severidad corresponde a su etiqueta y para lograr esto, se suma la probabilidad de 90 de subseñales de ser clasificadas en cada nivel de severidad y para resumir este proceso se puede observar la Figura 23. En otras palabras, la probabilidad máxima sumada es 90 y la mínima es de 0 para cada clase. Esto se debe a que en Matlab la probabilidad para cada subseñal tenía el rango de 0 a 1. Al determinar cada 90 veces cuál de las etiquetas era la más probable según la suma realizada, se llegaba a clasificar a una señal original con su respectiva etiqueta. De este modo, la matriz de confusión que se obtuvo es de 9×9 con el valor máximo de etiquetas correctamente clasificadas de 24 para el caso de los datos de entrenamiento y de 6 cuando se trata de los datos de entrenamiento.

Seguidamente, se realizó el **cálculo de métricas** para comparar el comportamiento y rendimiento de cada uno de los modelos. Se realizó el cálculo de precisión, exactitud, recall y F1-score para determinar si los resultados son los deseados, pero sobre todo para determinar si estos son los adecuados para poder establecer que existe una clasificación óptima con los modelos entrenados de los diferentes niveles de severidad de diente roto. Cada matriz de confusión generada corresponde a su respectivo modelo entrenado, por eso se puede definir que modelo género la matriz de confusión con mejores métricas calculadas. Con el fin explicar más sobre las métricas utilizadas presenta la Figura 24 donde se explica cómo a partir de una matriz de confusión se calcula la exactitud, precisión, recall y F1-score mediante sus respectivas fórmulas, dependiendo de los falsos positivos y negativos y también de los verdaderos positivos y negativos.

Una vez calculadas las matrices de confusión para todos los modelos generados, se realizó el **análisis de los mejores resultados obtenidos con las métricas**, donde se determinó el rendimiento de cada uno de los modelos entrenados según sus parámetros ajustados. Debido a que se generaron más de 7000 modelos, se facilitó su análisis al comparar los resultados de sus métricas. En el caso de KNN, de los 10 modelos generados por cada valor de k se escogió solamente uno, el cual tenía el mejor desempeño. De esta manera se redujo significativamente el proceso para definir qué modelo funciona mejor y entre los modelos que quedaban se comparan sus parámetros y resultados en relación con las métricas. Para RF se realizó lo mismo, pero según el valor de número de árboles dentro del bosque. Mediante valores obtenidos, sobre todo con la precisión y exactitud al clasificar, es posible saber si los modelos generados son

Figura 24

Cálculo de métricas mediante la matriz de confusión.



Nota: *Cálculo de métricas dentro del proyecto y matriz de confusión (Winn y cols., 2015).*

apropiados para desarrollar el diagnóstico del nivel de severidad de fallo de diente roto en engranajes rectos. Las métricas mencionas son los únicos valores disponibles y calculados para determinar el rendimiento de cada modelo, así que todo lo relacionado con los resultados y lo concluido en este trabajo se basa especialmente en la exactitud, precisión, recall y F1-Score.

Debido a la gran cantidad de resultados obtenidos, estos fueron resumidos en tablas y gráficos 3D (Véase sección) y para esto, en el caso de KNN se determinó el mejor modelo por número de k vecinos en vez de presentar los 10 modelos generados y en el caso de RF se determinó el mejor según su cantidad de árboles usando un determinado número de atributos (grupos de atributos obtenido por el *Ranking mencionado*). Al tener los resultados más relevantes entre varios modelos es posible **determinar el mejor modelo con sus respectivos parámetros** y esto es posible definir al encontrar los porcentajes más altos en relación con las métricas calculadas, especialmente la precisión y exactitud de clasificación del nivel de severidad de diente de fallo. Finalizados todos los procesos presentados en el diagrama de flujo, se puede definir qué modelo trabaja mejor y qué atributos se usaron durante el entrenamiento para determinar el nivel de severidad del fallo de diente roto, que es el enfoque principal del trabajo de titulación.

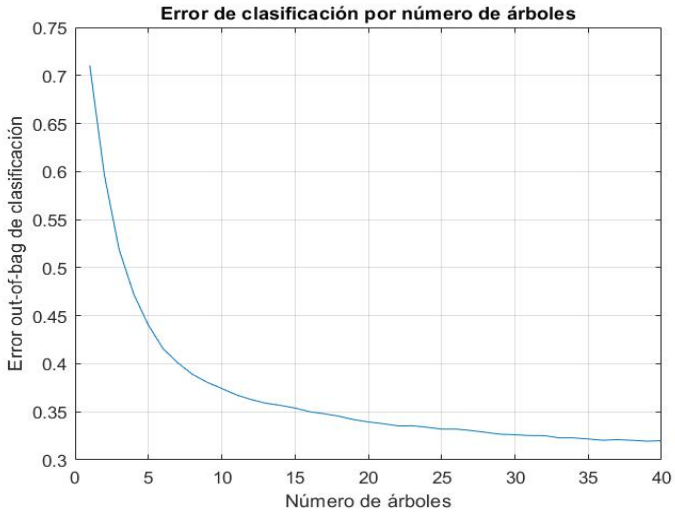
6.2.3. *Ranking* de atributos de Poincaré según Random Forest

Con el fin de ampliar los métodos de *Ranking* disponibles porque en un principio estos eran solamente: ReliefF y CDbws, se obtuvo un nuevo orden de *Ranking* del algoritmo de Random Forest. Los métodos de *Ranking* de atributos según su importancia. Para realizar el *Ranking* de atributos se utilizó un modelo de RF generado al seguir los pasos mostrados anteriormente en la sección 6.2.2. El proceso es muy similar porque es el mismo algoritmo RF usado para generar un modelo de clasificación que permitiera obtener métricas deseadas solamente con los datos de prueba. Los datos de entrenamiento se usaron tanto para el entrenamiento del algoritmo y para el cálculo de métricas mediante de la matriz de confusión generada. El objetivo básicamente era encontrar los parámetros de los modelos que generaron las mejores métricas y requieran de un mínimo de procesamiento para la computadora al implementar el algoritmo. Si los resultados obtenidos daban las métricas deseadas al clasificar los datos de prueba, se mantenían los parámetros establecidos y a partir de funciones de Matlab y el modelo generado se obtenía el *Ranking* de atributos según Random Forest. Para definir los parámetros de los modelos que tuvieron el mejor rendimiento, se decidió calcular el error Out-Of-Bag (OBB) según la cantidad de árboles generados dentro del bosque. Esto requería de que el error OBB calculado llegue a converger, es decir, que no haya cambio alguno o sino cambios mínimos en los valores del error después de cierta cantidad de árboles. Durante este proceso, se definieron 40 árboles como parámetro para la clasificación de atributos (Véase Figura 25) debido a que el error disminuye a medida que se ingresan más árboles dentro del bosque.

De este modo se obtuvo el *Ranking* de atributos de Poincaré y en la Figura 26 se puede observar en la tabla del lado derecho los diferentes porcentajes de importancia para cada uno de los atributos de Poincaré según Random Forest. A partir de este análisis se determinó que el valor de árboles seleccionados es de 40 para que empiece a converger el error mostrado en la Figura 25. Mediante esta imagen se puede concluir que los atributos más importancia son SD2C, Convex Hull y SD2. Para determinar si los atributos seleccionados por Random Forest son adecuados para KNN y Random Forest se debe clasificar los datos de prueba y con base en las métricas obtenidas es posible indicar el rendimiento de estos atributos y al mismo tiempo definir el mejor método de *Ranking* de atributos.

Figura 25

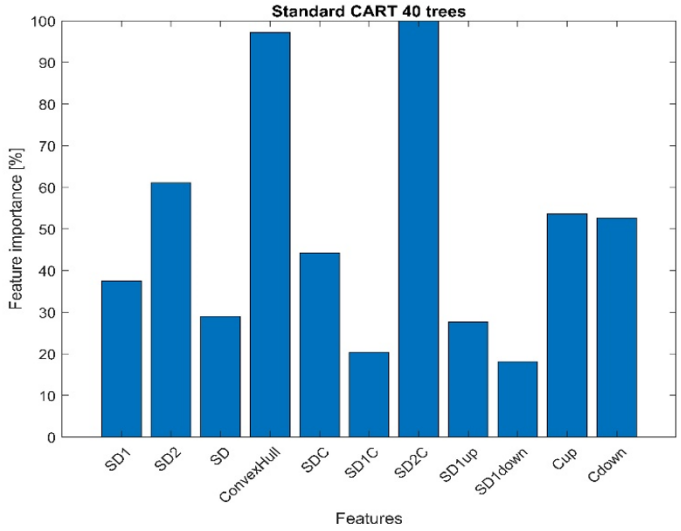
Error OBB según cantidad de árboles generados para generar el Ranking de atributos.



Nota: *En la Figura se puede observar que a partir de 40 árboles el erro OBB disminuye.*

Figura 26

Comparación de importancia y atributos para entrenamiento y prueba según método de Ranking



Feature	Importance
SD1	37,496%
SD1up	61,050%
SD1C	29,008%
SD1down	97,205%
SD2C	44,242%
SD2	20,3905%
SD	100%
SDC	27,644%
Convex Hull	18,117
Cup	53,621%
Cdown	52,657%

Nota: Importancia de atributos según el *Ranking* realizado por Random Forest.

7. Resultados

En este documento se estableció como objetivo la determinación del nivel de severidad de fallo de diente roto en una caja de engranajes a partir la señal de par eléctrico y de sus atributos de Poincaré. Se pudo realizar este objetivo debido a los resultados obtenidos por las métricas calculadas a partir de la clasificación de los diferentes modelos. De este modo, todo este capítulo está dedicado al resumen y la presentación de los mejores resultados al usar KNN y RF para la generación de modelos computacionales. El enfoque de los resultados está ligado a la determinación de los mejores parámetros, atributos y modelos. En primer lugar, se muestran los mejores resultados con KNN y RF para los datos de entrenamiento y prueba, para después hacer una comparación entre los dos algoritmos y finalmente se busca definir cuál de ellos funciona mejor al clasificar datos de prueba y bajo qué condiciones. Por otro lado, para determinar la influencia de atributos de Poincaré en la clasificación se seleccionaron los modelos que obtuvieron los mejores rendimientos con base en las métricas calculadas y una vez definidos los parámetros y los atributos usados en dichos modelos, se comparan por método de *Ranking*, ya sea CDbw, Random Forest y ReliefF con el fin de poder establecer que influencia tienen los atributos al clasificar.

7.1. Resultados para K-Nearest-Forest (KNN)

En esta sección se presenta un resumen general de los resultados de los modelos generados a partir de KNN mediante tablas, diagramas de barras y gráficas 3D. En primer lugar, se muestran los resultados de los mejores modelos con sus determinados parámetros y atributos. Después de definir los mejores modelos, se determina su rendimiento al ingresar los datos de prueba a los modelos y se compara los resultados de precisión, exactitud, entre otros, con los resultados obtenidos mediante los datos entrenamiento. Finalmente, se presenta los mejores resultados de clasificación al ingresar datos de prueba a los modelos ya entrenados. Un aspecto que se debe mencionar es que a pesar de haber definido los mejores modelos al clasificar los datos de entrenamiento, estos cambian cuando se aplica los datos de prueba. De este modo se definen los que tienen un mayor desempeño para clasificar datos de prueba. Adicionalmente, se compara los atributos usados en los mejores modelos computacionales generados por KNN dependiendo del método de *Ranking,* ya sea CDbw, Random Forest y ReliefF.

7.1.1. Resultados con KNN al clasificar los datos de entrenamiento.

En la Tabla 6 se muestra los mejores resultados obtenidos con CDbw, ReliefF y Random Forest y las diferentes distancias usadas para el algoritmo KNN. Se puede observar que el mejor resultado dentro de esta tabla fue con el *Ranking* de CDbw (grupo de 7 atributos) y ajustando los parámetros con la distancia Mahalanobis, el número de k vecinos de 46 y el modelo número 8 de la validación cruzada, pues su precisión es 95,88 % y su exactitud de 94,44 %. Por otro lado, se encuentra que el menor desempeño dentro los mejores resultados fue obtenido con el método de *Ranking* de Random Forest y sus parámetros fueron con la distancia Euclidiana, en donde se usaron 10 atributos, su número de k vecinos fue 46 y el número del modelo por validación cruzada fue el 3, es decir se utilizó es tercer modelo. Se reitera que por cada valor de k vecinos se generan 10 modelos para realzar la validación cruzada. Al final, el rendimiento de este modelo fue de 95,887 % para la precisión y 94,44 % para la exactitud. Se debe mencionar que también están los valores de recall o exhaustividad en español y el F1-score, estos dos valores no tienen mucha diferencia en relación con la exactitud para el caso de KNN.

Tabla 6

Tabla de mejores resultados con KNN.

Mejores resultados obtenidos con los diferentes métodos de ranqueo y diferentes distancias con KNN								
Método de ranqueo de atributos	**Distancia**	**Número de atributos**	**K-Vecinos**	**Número de modelo**	**Precisión**	**Exactitud**	**Recall**	**F1 Score**
CDbw	Euclidiana	5	46	9	93,048	91,203	91,203	90,703
	Coseno	11	46	6	93,480	91,666	91,666	91,116
	Mahalanobis	**7**	**46**	**8**	**95,877**	**94,444**	**94,444**	**93,897**
ReliefF	Euclidiana	10	46	1	92,899	91,203	91,203	90,703
	Coseno	9	46	8	94,227	91,666	91,666	91,324
	Mahalanobis	10	47	8	95,241	93,055	93,055	92,593
Random Forest	Euclidiana	10	46	3	92,899	91,203	91,203	90,703
	Coseno	7	47	2	93,480	91,666	91,666	91,095
	Mahalanobis	9	46	1	95,241	93,055	93,055	92,593

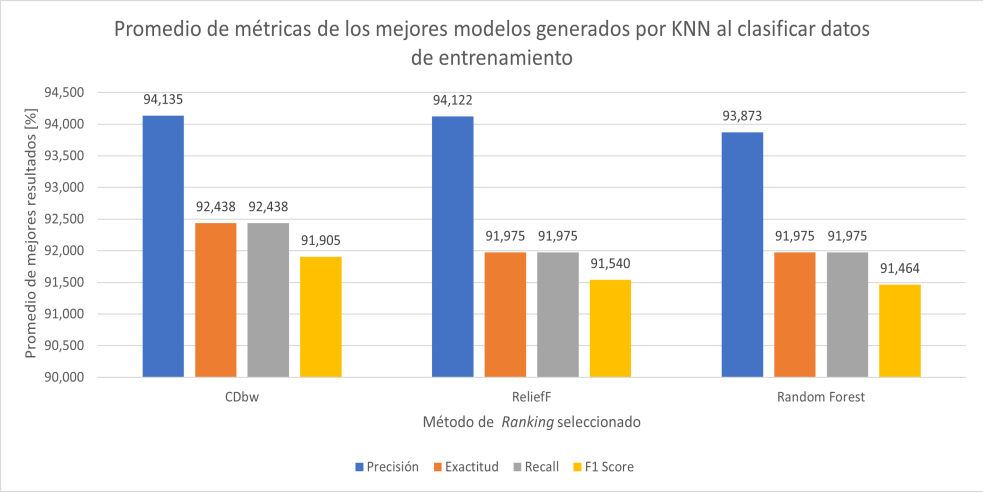
Nota: *En esta tabla se observa los mejores rendimientos de los modelos entrenados al clasificar los datos de entrenamiento usando diferentes distancias y métodos de Ranking.*

Con el fin de resumir la tabla presentada, se realizó un diagrama de barras mostrando los promedios de las métricas calculados para cada método de *Ranking* (Véase Figura 27), es decir, para realizar el diagrama se tomaron todos los mejores resultados de cada distancia mostrados

en la Tabla 6 y se promediaron según su *Ranking*. En el diagrama se muestran resultados para precisión, exactitud, recall y F1-score de CDbw (precisión 94,135 %, exactitud 92,438 %, recall 92,438 % y F1 Score 91,905 %), después de ReliefF (precisión 94,122 %, exactitud 91,975 %, recall 91,975 % y F1-score 91,540 %) y por último de Random Forest (precisión 93,873 %, exactitud 91,975 %, recall 91,975 % y F1-score 91,905 %).

Figura 27

Diagrama de barras de promedio de mejores métricas obtenidas con KNN.



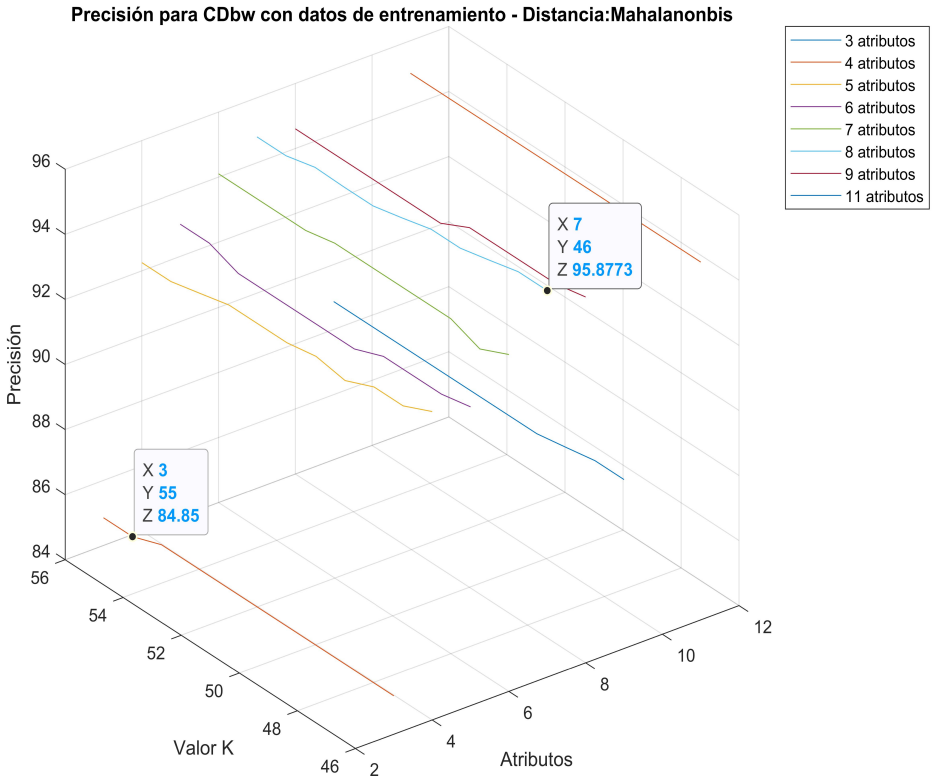
Nota: *Diagrama de barras de métricas obtenidas según método de Ranking.*

A partir de esto se concluye que los mejores modelos de clasificación fueron obtenidos al implementar los atributos del *Ranking* de CDbw y el *Ranking* que menor rendimiento tuvo fue Random Forest. Se debe mencionar que los modelos mostrados aquí hacen referencia mejores modelos entrenados al clasificar datos de entrenamientos y lo que se tiene comparar en este proyecto es si el rendimiento obtenido de estos modelos sigue siendo el deseado al implementar datos de prueba. En caso de haber modelos que tengan un mejor desempeño al clasificar datos de prueba, se debe realizar las conclusiones con base a esos modelos porque tienen la capacidad de identificar el nivel de severidad de diente roto a partir de nuevos datos. Del mismo modos, debido a que la cantidad de modelos generados era extensa, se requirió resumir los resultados mediante gráficas 3D. En la sección de Anexos se presentan los resultados del rendimiento de los mejores modelos generados. Estas gráficas muestran la precisión y exactitud respecto a la cantidad de k vecinos, la cantidad de atributos usados

dependiendo del método de *Ranking* y la distancia seleccionada para KNN, es decir, en total hay 18 gráficas 3D que abarcan más resultados que as tablas presentadas. A partir de los resultados obtenidos recientemente, el mejor desempeño se obtuvo con CDbw y la distancia de Mahalanobis, por eso a continuación en las Figuras 28 y 29 se muestran la precisión y exactitud de los mejores modelos obtenidos al usar dichos parámetros. Se puede observar en la Figura 28 que con 3 atributos y con $k = 55$ se llega una precisión de 84.85 %, la cual representa el peor rendimiento y en caso contrario a usar 7 atributos y con $k = 46$ se consigue una precisión de 95.877 %. Finalmente, en la Figura 29 se observa los resultados de exactitud de los mismos modelos mencionados para la precisión, donde el mejor modelo consigue una exactitud de 94.44 % y el de menor rendimiento tiene una exactitud de 78,703 %.

Figura 28

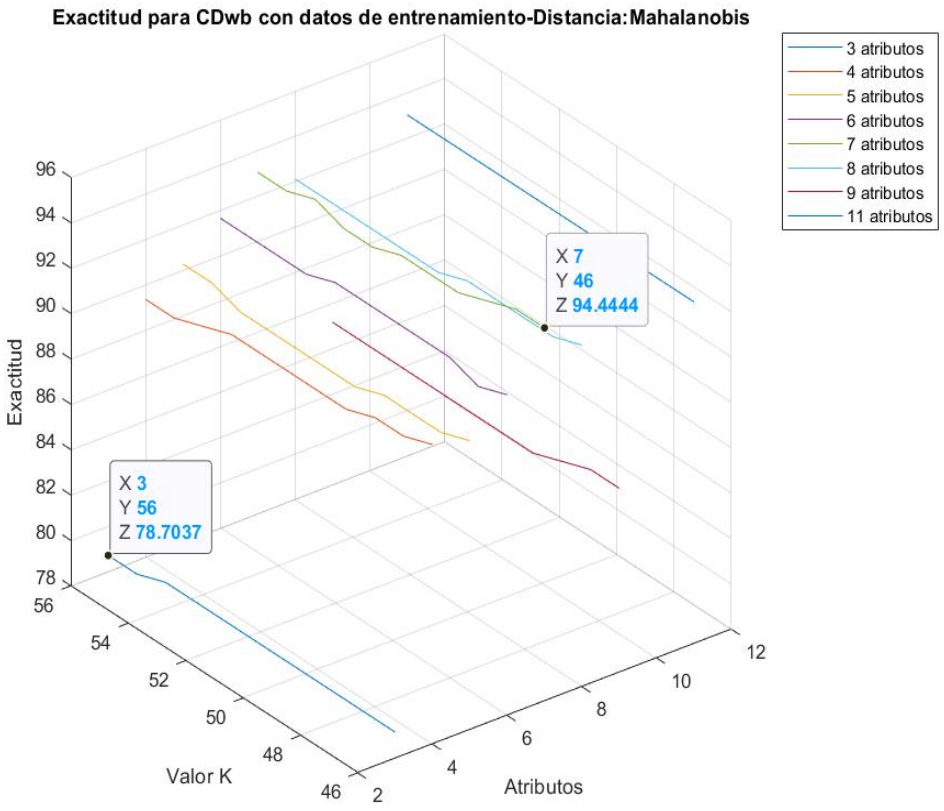
Precisión con CDbw usando la distancia Mahalanobis.



Nota: *Gráfica 3D de precisión con CDbw usando la distancia Mahalanobis.*

Figura 29

Exactitud CDbw.



Nota: *Gráfica 3D de exactitud con CDbw usando la distancia Mahalanobis.*

7.1.2. Resultados al ingresar los datos de prueba a los mejores modelos de KNN entrenados

Si bien los resultados mostrados indican que CDbw y la distancia Mahalanobis son los mejores parámetros para entrenar al algoritmo KNN, esto se debe poner a prueba al ingresar nuevos datos al algoritmo ya entrenado. En esta parte del documento se determinará el rendimiento real de KNN usando sus mejores modelos. En la Tabla 7 se comparan los resultados de mejores modelos según la clasificación realizada con los datos de entrenamiento probados con el ingreso de datos de prueba. En esta Tabla se comparan solamente la precisión y exactitud con el fin de no repetir datos y generar una perspectiva global de las diferencias

al momento de clasificar datos de entrenamiento y prueba. Los mejores resultados fueron obtenidos con CDbw tanto en la clasificación de los datos de entrenamiento como con los datos de prueba. Por lo general, los mejores resultados se obtiene al ingresar los datos de entrenamiento al modelo entrenado que al ingresar los datos de prueba y en este caso sucedió esto. De todos modos, esto no significa un sobre ajuste porque la diferencia entre las métricas obtenidas varía máximo hasta 4 % entre los datos de entrenamiento y prueba. Aunque el mismo modelo de CDbw con la distancia de Mahalanobis fue el mejor al momento de clasificar los datos de entrenamiento, hubo una disminución de 3 % en la precisión y de 4 % exactitud aproximadamente al ingresar los datos de prueba. Por otro lado, los mejores resultados de clasificación con los datos de prueba usando los mejores modelos fueron con la distancia Euclidiana usando 5 atributos seleccionados por CDbw, llegando a tener una precisión de 93,121 % y una exactitud de 90,74 %. Se puede resaltar que en este caso la precisión calculada con los datos de prueba fue mejor con los datos de entrenamiento; sin embargo, en relación con la exactitud, en este caso es menor con los datos de prueba (90,74 %) que con los datos de entrenamiento (91,203 %).

Tabla 7

Comparación de mejores modelos al clasificar datos de entrenamiento y prueba

Comparación de mejores modelos obtenidos durante el entrenamiento con datos de prueba							
Parámetros				Entrenamiento		Prueba	
Método de *Ranking*	Distancia	Número de modelo generado	Atributos	Precisión	Exactitud	Precisión	Exactitud
CDbw	Euclidiana	**5**	**9**	93,048	91,203	**93,121**	**90,740**
	Coseno	11	6	93,480	91,666	93,121	90,740
	Mahalanobis	**7**	**8**	**95,877**	**94,444**	92,857	90,740
ReliefF	Euclidiana	10	1	92,899	91,203	92,116	88,888
	Coseno	9	8	94,227	91,666	90,158	87,037
	Mahalanobis	10	8	95,241	93,055	92,857	90,740
Random Forest	Euclidiana	10	3	92,899	91,203	92,116	88,888
	Coseno	7	2	93,480	91,666	93,121	90,740
	Mahalanobis	9	1	95,241	93,055	92,857	90,740

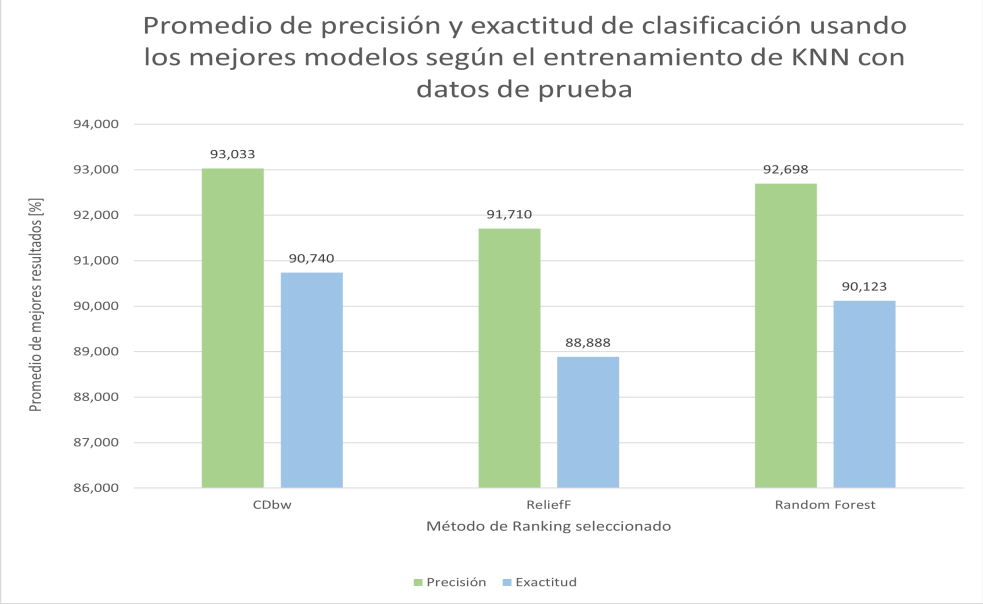
Nota: *Tabla de comparación del redimiento de los mejores modelos según KNN.*

En la Figura 30 se presenta un diagrama de barras y para este se calculó para cada uno de los métodos de *Ranking* el promedio de las métricas con los datos de prueba ingresados a los mejores modelos entrenados según el entrenamiento de KNN. Se puede observar que las

métricas más altas se obtuvieron con CDbw (precisión 93,033 % y exactitud 90,74 %), después con Random Forest (precisión 92,698 % y exactitud 90,123 %) y por último ReliefF (precisión 91,71 % y exactitud 88,884 %). De este modo se establece que para KNN el mejor método de *Ranking* para obtener las métricas adecuadas al clasificar se obtiene con CDbw.

Figura 30

Diagrama de barras del promedio de precisión y exactitud de clasificación con mejores modelos de KNN usando datos de prueba.



Nota: *Promedio de precisión y exactitud de clasificación usando los mejores modelos según el entrenamiento de KNN con datos de prueba.*

7.1.3. Resultados con KNN al clasificar los datos de prueba

En la Tabla 8 se indica los mejores resultados obtenidos con datos de prueba de todos los modelos generados por KNN y por método de *Ranking* de atributos. Nuevamente, se puede definir que los mejores parámetros para clasificar con KNN son: atributos clasificados por CDbw y con la distancia Mahalanobis, pues la precisión es de 94,444 % y la exactitud 92,592 %. Se debe menciona que lo mostrado en la tabla son los modelos que mejor desempeño tuvieron al clasificar nuevos datos y sus ajustes son diferentes al de los mejores modelos según el entrenamiento.

Tabla 8

Tabla de los mejores resutlados clasificación de datos de prueba con KNN.

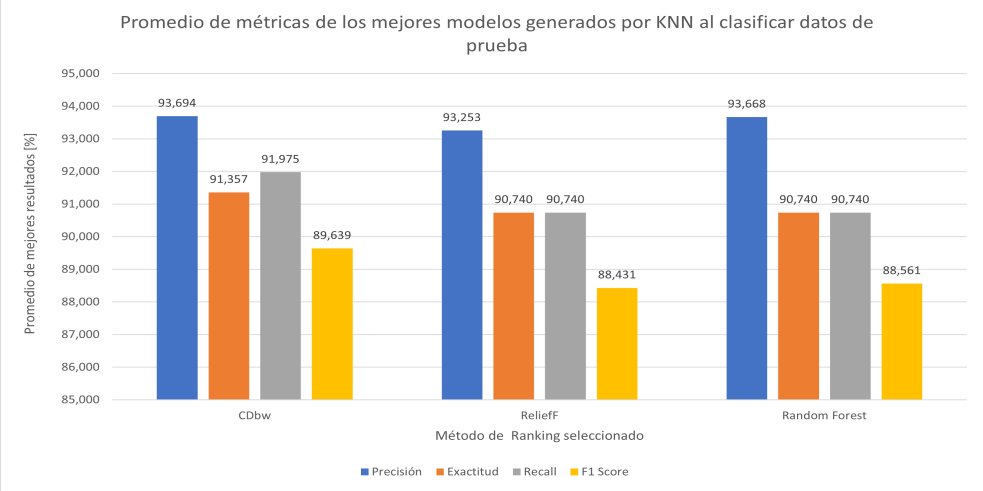
Mejores resultados de clasificación con KNN								
Método de ranqueo de atributos	**Distancia**	**Número de atributos**	**K-Vecinos**	**Número de modelo**	**Precisión**	**Exactitud**	**Recall**	**F1 Score**
CDbw	Euclidiana	6	48	4	93,518	90,740	90,740	88,692
	Coseno	10	46	1	93,121	90,740	90,740	88,692
	Mahalanobis	**6**	**48**	**6**	**94,444**	**92,592**	**94,444**	**91,533**
ReliefF	Euclidiana	11	47	9	93,121	90,740	90,740	88,300
	Coseno	11	46	2	93,121	90,740	90,740	88,300
	Mahalanobis	10	46	9	93,518	90,740	90,740	88,692
Random Forest	Euclidiana	4	54	5	93,968	90,740	90,740	88,692
	Coseno	6	46	4	93,518	90,740	90,740	88,496
	Mahalanobis	4	46	1	93,518	90,740	90,740	88,496

Nota: *Promedio de precisión y exactitud de clasificación usando los mejores modelos según el entrenamiento de KNN al clasificar datos de prueba.*

No obstante, en la Figura 31 los resultados obtenidos al clasificar los datos de prueba, en donde según la gráfica de barras, el mejor método de *Ranking* es CDbw (precisión 93,694 % y exactitud 91,357 %), después está Random Forest (precisión 93,668 % y exactitud 90,740 %) y por último están los resultados más bajos que fueron obtenidos con ReliefF (precisión 93,253 % y exactitud 90,740 %). A partir de estas métricas es posible notar diferencias con los resultados obtenidos con los datos de entrenamiento en la sección 7.1.1. El desempeño de los modelos entrenados por lo general es menor al usar datos de prueba, pero una característica que comparten los resultados de clasificación de datos de entrenamiento y prueba es que el mejor rendimiento fue obtenido con CDbw y con la distancia de Mahalanobis, aun así otros ajustes de parámetros como el valor de k y la cantidad de atributos fue diferente para los dos. A partir de esto es posible decir que los datos ingresados según cada método de *Ranking* a los modelos entrenados influyen en los resultados al clasificar y aunque determinados modelos sean mejores al clasificar los datos de entrenamiento, esto no significa necesariamente que estos mismos sean los mejores al clasificar los datos de prueba. Al final, lo importante es que el modelo tenga la capacidad de determinar el nivel de fallo de diente roto al ingresar nuevos datos y el enfoque de este proyecto es hacia estos resultados.

Figura 31

Diagrama de barras del promedio



Nota: *Diagrama de barras del promedio de los mejores resultados de KNN al calsificar datos de prueba.*

7.1.4. Comparación de atributos usados en los mejores modelos de clasificación KNN de datos de entrenamiento y prueba

La tabla en la Figura 9 indica los atributos usados en los mejores modelos de KNN al momento de clasificar tanto datos de entrenamiento como de prueba. Los mejores resultados fueron obtenidos con modelos entrenados con el *Ranking* de CDbw y para comparar los atributos usados en los mejores modelos según CDbw se observa que para los datos de entrenamiento se necesitan 7 atributos y para clasificar los datos de prueba se necesitan 6 (SD, Cup, SD2, SD2C, SD1up y Convex Hull) atributos. Debido a que se mantiene el método de *Ranking* de CDbw, los atributos usados son exactamente los mismos que en entrenamiento, pero para los datos de prueba no se incluye al atributo SD1. Para los mejores modelos obtenidos al implementar el *Ranking* de ReliefF se usaron 10 atributos tanto para clasificar los datos de prueba y entrenamiento en los mejores modelos generados a partir de ReliefF. Estos 10 atributos son: SD1, SD1up, SD1C, SD1down, SD2C, SD2, SD, SDC, Convex Hull y Cup. Por otro lado, con Random Forest se usaron 10 atributos para clasificar los datos de entrenamiento y 4 atributos para los datos de prueba.

Se puede concluir a partir de la Tabla 9 que al clasificar datos de entrenamiento, el *Ranking* de CDbw usa menos características que el resto de modelos en comparación de Random Forest y ReliefF. Los tributos que no comparten estos dos métodos de *Ranking* son SD1down y Cdown, pero 9 de 10 atributos usados de estos dos métodos son iguales. Ahora, en relación con modelos que mejores resultados obtuvieron al clasificar los datos de prueba, se puede definir que los atributos más importantes y comunes en todos los métodos *Ranking* son: SD2C, Convex Hull, SD2 y Cup. De esta manera, es posible decir que estos atributos se deben tener en consideración para obtener buenos resultados al clasificar los diferentes niveles de severidad de fallo. Debido a que se determinó que CDbw es el mejor método para ordenar atributos según su importancia para mejorar, aún más el rendimiento del modelo se pueden incluir los atributos SD y SD1up.

Tabla 9

Comparación de importancia y atributos para entrenamiento y prueba según método de Ranking.

Atributos usados en los mejores modelos al clasificar datos de entrenamiento y prueba						
Método de Ranking	CDbw		ReliefF		Random Forest	
Tipo de datos	**Entrenamiento**	**Prueba**	**Entrenamiento**	**Prueba**	**Entrenamiento**	**Prueba**
Nombres de atributos	SD	SD	SD1	SD1	SD2C	SD2C
	SD1	Cup	SD1up	SD1up	Convex Hull	Convex Hull
	Cup	SD2	SD1C	SD1C	SD2	SD2
	SD2	SD2C	SD1down	SD1down	Cup	Cup
	SD2C	SD1up	SD2C	SD2C	Cdown	-
	SD1up	Convex Hull	SD2	SD2	SDC	-
	Convex Hull	-	SD	SD	SD1	-
	-	-	SDC	SDC	SD	-
	-	-	Convex Hull	Convex Hull	SD1up	-
	-	-	Cup	Cup	SD1C	-
	-	-			-	-

Nota: *Comparación de atributos usados en los mejores modelos al clasificar datos de entrenamiento y prueba según método de Ranking.*

7.2. Resultados con Random Forest(RF)

En esta parte del capítulo se presentan los diferentes resultados de la clasificación de los datos de entrenamiento y de prueba para definir cuál modelo tiene el rendimiento más alto y

la influencia de los atributos dentro de la clasificación. En este caso, se determina el mejor modelo con relación a las métricas calculas y según los parámetros que este posea es posible establecer el ajuste adecuado de los mismos.

7.2.1. Resultados con Random Forest al clasificar los datos de entrenamiento

Una vez obtenidos todos los modelos entrenados de Random Forest con sus respectivos parámetros (cantidad de árboles y divisiones), así como los diferentes grupos de atributos seleccionados, ya sea por CDbw, Random Forest y ReliefF, se realizó la clasificación de los datos de entrenamiento, donde los resultados obtenidos para todas las métricas fueron de 100 % para todos los casos. El resumen de lo mencionado se puede observar en la Tabla 10 que indica el promedio de los resultados de las métricas obtenidas con los distintos métodos de *Ranking*. En conclusión, todos los modelos funcionan igual de bien al clasificar los datos de entrenamiento, independientemente de los parámetros seleccionados, así como con la cantidad de árboles y cantidad de atributos usados cuando se utiliza una división de árboles de un rango sobre 1900 hasta 5000, esto para evitar la generación de árboles profundos dentro del bosque. En la Figura 32 se observa el promedio de todas las métricas de los mejores modelos obtenidos al clasificar los datos de entrenamiento. En este diagrama de barras se observa que todas las métricas tienen 100 %.

Tabla 10

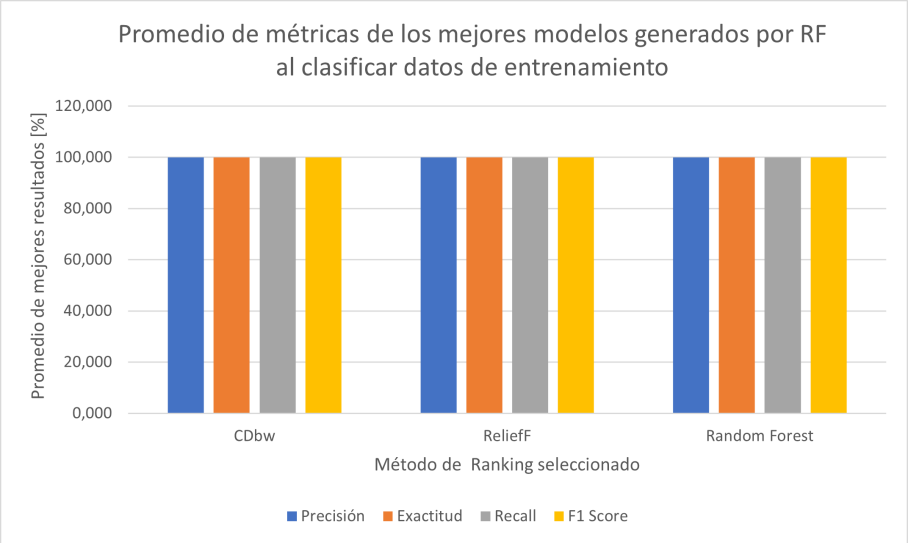
Tabla de resultados de métricas usando Random Forest con datos de entrenamiento con todos los grupos de atributos seleccionados por CDbw, Random Forest y ReliefF.

Resultados de Random Forest con datos de entrenamiento con grupos de 3 a 11 atributos seleccionados por CDbw, Random Forest y ReliefF				
Árboles	**Precisión**	**Exactitud**	**Recall**	**F1 Score**
5	100	100	100	100
10	100	100	100	100
15	100	100	100	100
20	100	100	100	100
25	100	100	100	100
30	100	100	100	100
35	100	100	100	100
40	100	100	100	100
45	100	100	100	100
50	100	100	100	100
55	100	100	100	100
60	100	100	100	100
65	100	100	100	100

Nota: *En la tabla se observa que todo los modelos tuvieron el mismo rendimeinto.*

Figura 32

Gráfica de barras del promedio de la precisión y exactitud de clasificación de RF con datos de entrenamiento.



Nota: *Diagrama de barras de Random Forest al clasificar datos de entrenamiento con todos los métodos de Ranking.*

7.2.2. Resultados con Random Forest al clasificar los datos de prueba

Al clasificar los datos de entrenamiento, no se determinaron qué modelos específicamente de Random Forest eran los mejores como en el caso de KNN, puesto que todos tenían el mismo rendimiento, el cual era el máximo valor para todos que representa el 100 %. Un aspecto a considerar al momento de obtener los resultados para los datos de prueba es el número de divisiones del árbol. Este valor se determinó que a partir de 1900 divisiones debido a que los resultados de clasificación de los datos de entrenamiento no mejoraban, pero la calificación de los prueba sí podían mejorar con más divisiones de árbol. Con base en ese análisis se tomó la decisión de realizar divisiones mayores a 1900, pero que al menos se encuentre en un rango máximo de hasta 5000 divisiones. De este modo, el parámetro escogido en este trabajo fue de 5000 como número máximo de divisiones de árbol.

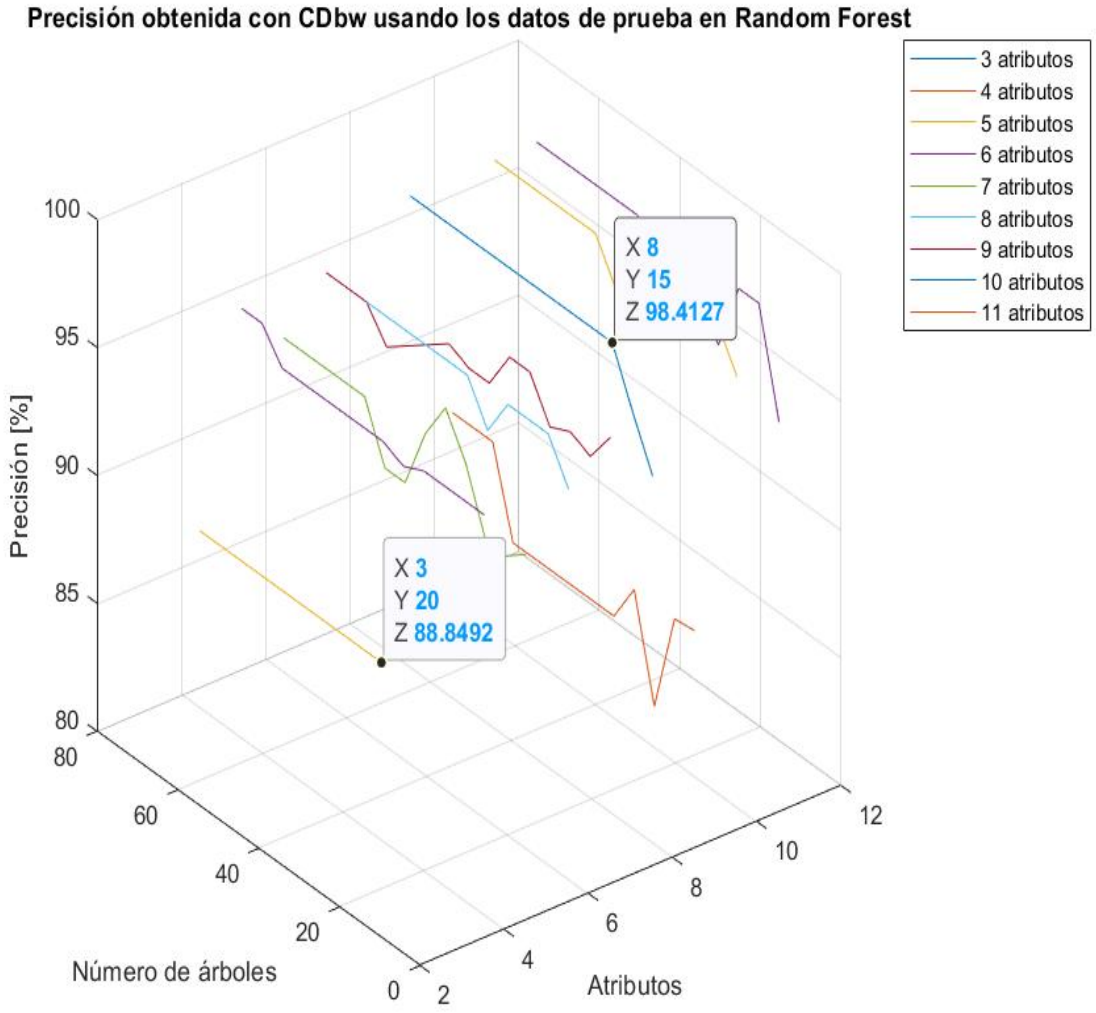
La clasificación de los datos de prueba se presentan las Figuras 33 y 34 que son gráficas 3D en donde los ejes indican la cantidad de atributos usados, el número de árboles en el

bosque y el porcentaje de la correspondiente métrica. Se realizaron varias de estas gráficas que pueden ser observadas con mayor detalle en la parte de Anexos en la sección . En cuanto a las gráficas mostradas, aquí resumen todos los resultados obtenidos con Random Forest usando determinada cantidad de atributos seleccionados por CDbw y cantidad de árboles dentro del bosque aleatorio. Como se observa en la Figura 33 el mejor resultado de precisión con datos de prueba fue 98,4127 % y en cuanto al resultado más bajo fue de 88,849 %. En cambio, en la Figura 34 se muestra la exactitud máxima de 98,1481 % y la mínima de 87,037 %. Se sabe que hay 3 diferentes métodos de *Ranking*; sin embargo, se indican los resultados solamente de CDbw debido a que este método tiene un buen rendimiento al clasificar datos de prueba (Véase Figura 35) en comparación de los otros métodos. Esto se explica con mayor detalle en el siguiente párrafo.

En las Tablas 11, 12 y 13 se muestran los mejores resultados para cada método de *Ranking* según la cantidad de árboles del bosque aleatorio. Debido a que para cada grupo de atributos se crecieron de 5 hasta 65 árboles, solo los resultados más altos en cuanto a las métricas fueron seleccionados y también se decidió promediar estos resultados para poder realizar el diagrama de la Figura 35 que específicamente es un diagrama de barras. Mediante este diagrama se llega a determinar que al momento de clasificar los datos de prueba usando los atributos seleccionados por CDbw (precisión 96,019 % y exactitud 95,063 %) se obtiene el rendimiento más alto en comparación con ReliefF (precisión 95,212 % y exactitud 94,033 %) y Random Forest (precisión 91,748 % y exactitud 90,737 %). CDbw es el mejor método de *Ranking* para aplicar este algoritmo debido a su alto rendimiento y bajo procesamiento, el cual consta del uso de 8 atributos y 15 árboles. Un aspecto que se puede mencionar es que en la Figura 35 se observa que en promedio ReliefF requiere de 16 árboles para obtener los mejores resultados, pero tanto Random Forest y ReliefF (24 árboles para los dos casos) necesitan una cantidad más alta. Aun así, más árboles no necesariamente significa mejores resultados y basándose en los promedios calculados para el algoritmo de Random Forest se sugiere que al usar determinados métodos de *Ranking*, la cantidad de árboles se debe considerar así como el número de divisiones de árbol. De esta manera, aunque ReliefF puede llegar a dar los mismos resultados que CDbw solo necesitar 5 árboles, la cantidad de atributos que 9 atributos. Debido a que se busca generar árboles aleatorios 5 se considera un número bajo esto.

Figura 33

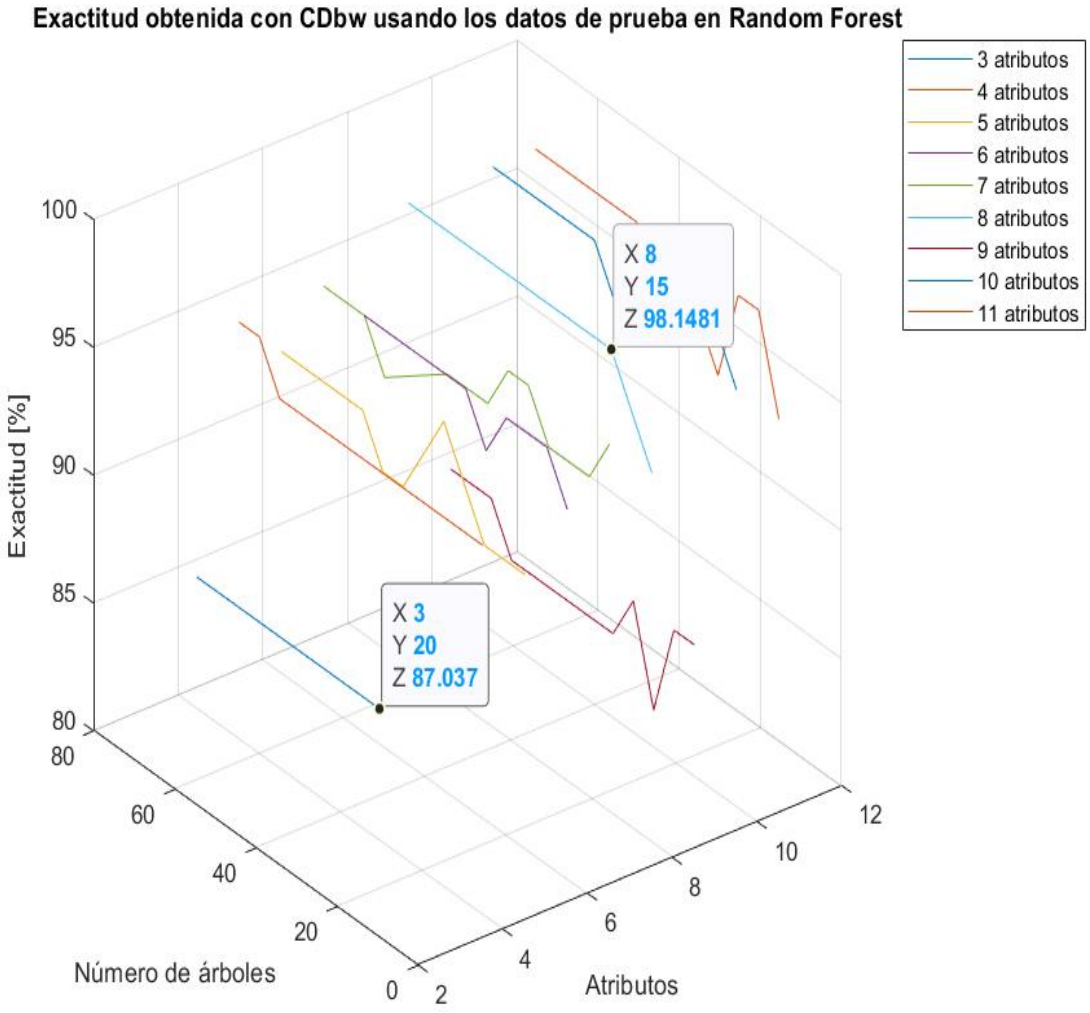
Precisión obtenida con CDbw usando los datos de prueba en Random Forest.



Nota: *Gráficas 3D de precisión obtenida con CDbw usando los datos de prueba en Random Forest.*

Figura 34

Exactitud obtenida con CDbw usando los datos de prueba en Random Forest.



Exactitud: *Gráficas 3D de precisión obtenida con CDbw usando los datos de prueba en Random Forest.*

Tabla 11

Mejores resultados con Random Forest usando CDbw.

Mejores resultados de entrenamiento con Random Forest usando CDbw					
Atributos	**Árboles**	**Precisión**	**Exactitud**	**Recall**	**F1 Score**
3	5	92,381	88,888	88,888	86,957
4	60	96,825	96,296	96,296	96,154
5	25	96,825	96,296	96,296	96,154
6	10	96,825	96,296	96,296	96,154
7	25	96,825	96,296	96,296	95,488
8	**15**	**98,413**	**98,148**	**98,148**	**98,039**
9	55	89,246	87,037	87,037	84,746
10	10	98,413	98,148	98,148	98,039
11	10	98,413	98,148	98,148	98,039
Promedio	23,889	96,0184	95,0614	95,0614	94,419

Nota: *Tabla de mejores resultados con Random Forest usando CDbw al clasificar datos de prueba.*

Tabla 12

Mejores resultados con Random Forest usando Random Forest.

Mejores resultados de entrenamiento con Random Forest usando Random Forest					
Atributos	**Árboles**	**Precisión**	**Exactitud**	**Recall**	**F1 Score**
3	10	84,418	81,481	81,481	79,681
4	5	93,519	90,741	90,741	88,692
5	10	94,444	92,593	92,593	90,703
6	45	96,825	96,296	96,296	96,154
7	5	96,825	96,296	96,296	96,154
8	**25**	**98,413**	**98,148**	**98,148**	**98,039**
9	65	95,635	94,444	94,444	93,897
10	45	98,413	98,148	98,148	98,039
11	10	98,413	98,148	98,148	98,039
Promedio	24,444	95,212	94,033	94,033	93,266

Nota: *Tabla de mejores resultados con Random Forest usando Random Forest al clasificar datos de prueba.*

Tabla 13

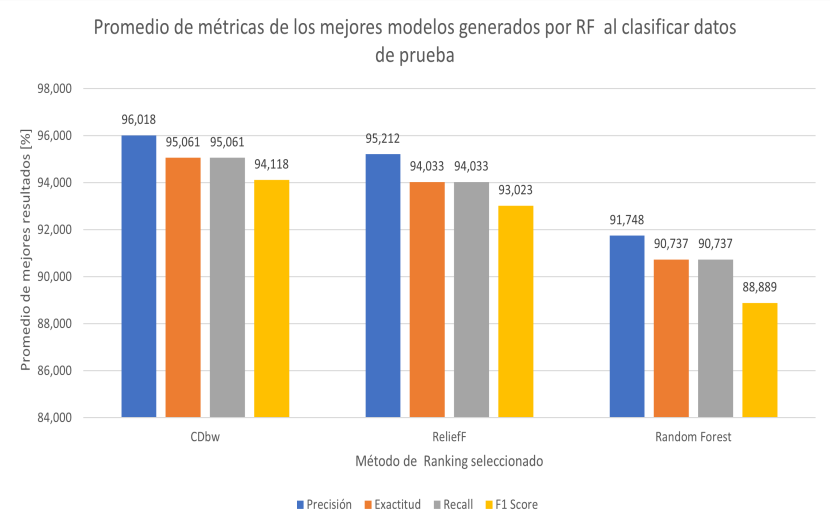
Tabla de mejores resultados con Random Forest usando ReliefF.

Mejores resultados de entrenamiento con Random Forest usando ReliefF					
Atributos	**Árboles**	**Precisión**	**Exactitud**	**Recall**	**F1 Score**
3	10	87,778	85,185	85,185	83,6820
4	25	89,074	87,037	87,037	85,6531
5	20	85,159	85,185	85,185	81,7996
6	20	88,624	87,037	87,037	85,6531
7	5	90,238	88,889	88,889	87,7193
8	10	89,616	88,889	88,889	87,7193
9	5	**98,413**	**98,113**	**98,113**	**98,0392**
10	40	98,413	98,148	98,148	98,0392
11	10	98,413	98,148	98,148	98,0392
Promedio	16,111	91,748	90,737	90,737	88,8889

Nota: *Resultados con Random Forest usando ReliefF al clasificar datos de prueba.*

Figura 35

Diagrama de barras del promedio de las métricas de los mejores resultados obtenidos con Random Forest.



Nota: *Diagrama de barras del promedio de las métricas de los mejores modelos entrenados por Random Forest al clasificar los datos de prueba.*

7.2.3. Mejores resultados obtenidos al clasificar datos de prueba con Random Forest.

Como se logró ver anteriormente, los mejores resultados fueron obtenidos con CDbw y aunque la mayoría de sus atributos son compartidos tanto por ReliefF y Random Forest, CDbw no incluye atributos como S1down, SDC y Cdown, lo cual se puede observar en la Tabla 14. En esta Tabla se puede ver los atributos usados en los modelos RF que tuvieron mayor rendimiento al clasificar según su método *Rankig*. Al final, debido a que CDbw y Random Forest tuvieron mejores resultados que ReliefF (Véase Fig. 35), se menciona que el atributo SD1down no es compartido los otros dos métodos. A partir esta observación es posible decir que Cdown y Cup obtienen un mejor rendimiento que SD1down en comparación al usar SD1C y SD1down. Esto se puede notar claramente encontrando los atributos en común entre ReliefF y Random Forest. En todo caso se puede ver en la sección 7.2.2 los resultados de rendimiento obtenidos con estos métodos de *Ranking*.

Tabla 14

Tabla de tributos usados que obtuvieron mejor desempeño al clasificar según CDbw, Random Forest y ReliefF.

Método de Ranking	CDbw	ReliefF	Random Forest
Nombres de atributos	SD	SD1	SD2C
	SD1	SD1up	Convex Hull
	SD1C	SD1C	SD2
	Cup	SD1down	Cup
	SD2	SD2C	Cdown
	SD2C	SD2	SDC
	SD1up	SD	SD1
	Convex Hull	SDC	SD
	-	Convex Hull	-
	-	-	-
	-		-

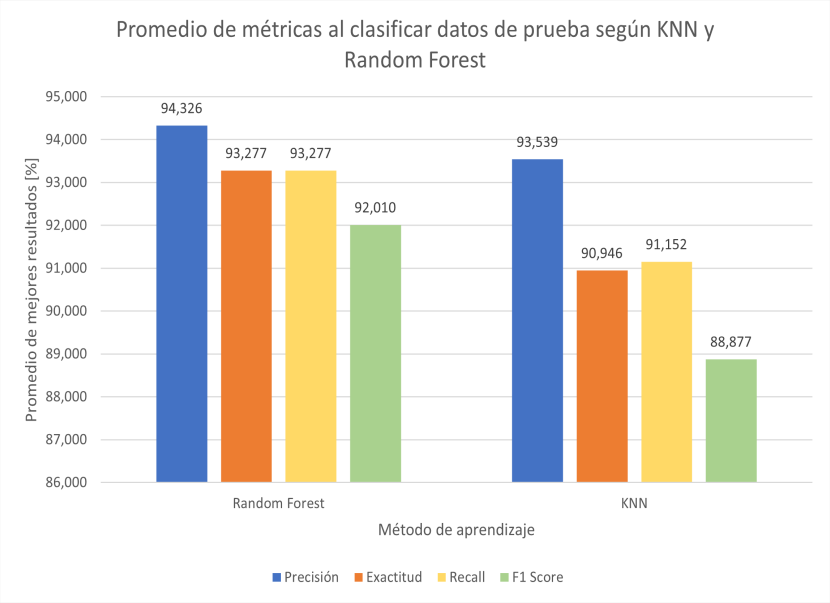
Nota: *Comparación de atributos usados en mejores modelos según RF.*

7.3. Resultados y comparación de mejores parámetros, atributos y modelos con KNN y RF.

Finalmente, se logró aplicar los algoritmos KNN y Random Forest para generar modelos con la capacidad de clasificar de nivel de severidad de fallo de rotura de diente en una caja de engranajes a partir de la señal del par eléctrico usando sus atributos de Poincaré. En la Figura 36 se puede observar el promedio de los mejores resultados de las métricas obtenidas tanto para KNN y Random Forest. Para realizar esta gráfica el enfoque fue hacia el rendimiento y no hacia los parámetros usados por los mejores modelos. En esta gráfica se puede observar que la diferencia entre KNN y RF para la precisión es de aproximadamente 0,8% y de 2% para la exactitud y recall, pero para F1-score se llega a casi un 3% de diferencia entre modelos generados al implementar estos dos algoritmos.

Figura 36

Comparación de los promedios de los mejores resultados entre Random Forest y K-Neartest-Neighbor.



Nota: *Comparación final en KNN y RF. Se observa que su rendimeinto es similar.*

De este modo, aunque los dos algoritmos, en general, según sus promedios, tienen resultados muy similares, el mejor algoritmo de clasificación fue Random Forest, puesto genero un modelo

computacional que alcanzó una precisión y exactitud al rededor de 98 % clasificando los datos de prueba. Este resultado se obtuvo de forma específica y usando determinados parámetros, por eso en caso de usar este modelo para clasificar nuevos datos, no va a obtener necesariamente este mismo rendimiento, pero lo más probable es que pueda diagnosticar el nivel de severidad de diente roto. Los resultados con Random Forest fueron los mejores y tienen un alto desempeño, pero que hay aspectos que mencionar. Por lo general, se suele asumir que cuando la exactitud de entrenamiento es del 100 % el algoritmo puede estar sobreajustado; no obstante, en este caso, se generó un código diferente para la clasificación pero igualmente válido para señales originales a partir de subseñales obtenidas al aumentar los datos disponibles (Véase sección 6.2.2). Por otro lado, aunque ReliefF y Random Forest obtuvieron resultados muy buenos, el mejor método de *Ranking* fue CDbw con la cantidad de 8 atributos y 15 árboles para llegar a la exactitud y precisión mencionadas. Por lo tanto, los mejores atributos son: SD, SD1, SD1C, Cup,SD2, SD2C, SD1up y Convex Hull. En cuanto a los resultados promediados, para RF se obtuvo una precisión de 94,326 % y exactitud de 93,277 % y para KNN se obtuvo una precisión 93,539 % y con una exactitud de 90,946 %. Mediante estos resultados es posible decir que en general cualquiera de estos dos algoritmos usados para la generación de modelos capaces de clasificar los diferentes nivele de fallo de diente roto en una caja de engranajes son buenas opciones, pero ligeramente RF supera a KNN.

8. Conclusiones

El Mantenimiento Predictivo permite determinar la condición de un equipo en servicio para estimar cuándo se debe realizar actividades de mantenimiento. Al aplicar técnicas del Mantenimiento Basado en la Condición (CBM) es posible evitar fallas de maquinaria rotativa que contribuye a la producción de la industria. En este caso, mediante el monitoreo de la señal de par eléctrico se puede detectar y diagnosticar el grado de rotura de diente en una caja de engranajes rectos usando algoritmos entrenados como KNN y RF para la generación de modelos computacionales. Para que el entrenamiento de estos algoritmos logré generar modelos capaces de clasificar adecuadamente los datos de prueba, es necesario escoger los parámetros óptimos dependiendo del algoritmo. Para este trabajo se usaron parámetros propios de los algoritmos (por ejemplo, el número de k vecinos para KNN y el número de árboles para RF) y, por otro lado, se usaron parámetros como la cantidad de atributos de Poincaré o grupos de atributos usados dependiendo del método de *Ranking* seleccionado, ya sea CDbw, Random Forest o ReliefF. Considerando estos parámetros y modificándolos

constantemente, fue posible calcular métricas de cada uno de los modelos generados al clasificar los datos de prueba y de entrenamiento. Las métricas como la precisión, exactitud, etc. fueron usadas para establecer qué parámetros, atributos y modelo es el mejor al clasificar el nivel de rotura de engranajes rectos. De este modo fue posible aplicar KNN y RF para generar modelos que pueden realizar el diagnóstico de una caja de engranajes a partir de los atributos de Poincaré calculados de la señal de par eléctrico del motor acoplado a dicha caja.

La base de datos de la señal de par eléctrico fue obtenida en un proyecto anterior de Ortega Lucero (2021) y después esta señal fue procesada para obtener los atributos de Poincaré usados en el desarrollado de un análisis exploratorio de Mejía (2022). A partir de estos trabajos fue posible complementar este trabajo de titulación. De todos modos, la forma de la cual se procesó la señal de par eléctrico y la obtención de los atributos de Poincaré son descritos brevemente en la metodología. Los atributos de Poincaré demuestran ser indicadores con buenos resultados al generar modelos para diagnosticar el estado de caja de engranajes, pero tomar solamente los atributos de Poincaré como parámetro a modificar no es suficiente para definir si un modelo es mejor que el otro porque los parámetros propios de cada algoritmo influyen en el rendimiento del modelo también. El desempeño de los modelos generados por los algoritmos KNN y RF al clasificar se basó en el análisis de resultados (métricas obtenidas) al ingresar los datos de prueba en los modelos ya entrenados. Al comparar estas métricas se definió el mejor algoritmo para la generación de modelos con alto regimiento y después se identificó los parámetros que el mejor modelo generado poseía. En el caso de este trabajo de titulación, el mejor algoritmo de clasificación fue Random Forest usando 8 atributos seleccionados por CDbw y usando 15 árboles. En cuanto a aspectos generales, al comparar la precisión de los diferentes modelos computacionales generados con KNN y RF, se estableció que RF con una precisión de 94,325 % es mejor que KNN con una precisión de 93,538 %. En promedio, los mejores modelos y resultados fueron obtenidos con RF pero la diferencia entre KNN y RF no es del todo significativa debido a las bajas variaciones de los valores de las métricas calculadas.

Por otro lado, con relación a los atributos de Poincaré se había determinado que tanto para KNN y Random Forest, el mejor método de *Ranking* es CDbw. Como se describió anteriormente, la importancia y cantidad de los atributos que son agrupados por el método de *Ranking* correspondiente influye en los resultados de clasificación. Al ser CDbw el mejor método de *Ranking*, los atributos SD2 y Cup influyen en la clasificación de los datos de prueba de forma más notoria en comparación de los otros atributos. En KNN se usaron, en general, más atributos, pero sus resultados en cuanto a las métricas no superan a Random Forest; sin embargo, no

difieren mucho. De esta manera, para conseguir resultados satisfactorios al diagnosticar el nivel de rotura de diente de una caja de engranajes usando los atributos de Poincaré de la señal de par eléctrico y los algoritmos como KNN y Random Forest, se debe seleccionar un método de *Ranking* apropiado y basándose en los diferentes grupos generados, estableció los parámetros propios del algoritmo que funcionan mejor. Los mejores parámetros son: la raíz cuadrada de la cantidad de los datos de entrenamiento, así como el número de vecinos para KNN y para Random Forest no hay cálculos específicos para establecer los mejores parámetros, pero para este trabajo, si se pudo determinar que a partir de 40 árboles los resultados de clasificación ya no mejoraban, es decir, convergían justamente como se determinaba en el error OBB de la sección.

Finalmente, se logró cumplir con todos los objetivos establecidos en un principio, pero principalmente fue posible diagnosticar diferentes niveles de severidad de diente roto (desde P1 hasta P9) en una caja de engranajes a partir de los atributos de Poincaré de la señal del par eléctrico usando algoritmos de aprendizaje automático supervisados como KNN y RF con un buen rendimiento al momento de clasificar.

9. Recomendaciones

Como recomendaciones sobre el trabajo realizado se debe mencionar que existieron ciertas observaciones al momento de trabajar con los algoritmos KNN y Random Forest. Para el primer caso, la distancia Mahalanobis requería de una matriz de covarianza obtenida a partir de los datos de entrenamiento y para todos los atributos se generó dicha matriz, pero al usar los 10 de los atributos seleccionados por CDbw y Random Forest, la matriz generada no era la apropiada para que Matlab generase el modelo entrenado, por eso no existen resultados para KNN a usar los parámetros mencionados. Por otra parte, en Random Forest se usó al inicio el valor de la mitad de la cantidad de datos de entrenamiento para el número de divisiones de los árboles del bosque aleatorio, sin embargo, esto generaba árboles muy profundos y mayor tiempo de procesamiento, por eso se determinó 5000 divisiones como número máximo.

Tomando en cuenta lo mencionado en el párrafo anterior, es posible trabajar con los mismos algoritmos, pero cambiando otros tipos de parámetros, por ejemplos para KNN usar otro tipo de distancia o generar una matriz de covarianza apropiada con los mismos datos y para Random Forest, se puede cambiar la cantidad de padres, hojas, etc. También es posible usar otros métodos de *Ranking* de atributos y probablemente se pueda superar los

resultados obtenidos por los modelos al usar CDbw, incluso se podría considerar usar otro tipo de atributos además de los atributos de Poincaré. Debido a que la señal del par eléctrico es sensible a la severidad de diente roto en una caja de engranajes rectos, igualmente se pueden aplicar otros algoritmos no supervisados como Support Vector Machines. Hay muchas variables que se pueden considerar al momento de realizar trabajos futuros de clasificación de fallo de la señal del par eléctrico; sin embargo, son algunas de las posibilidades a considerar.

Referencias

Aggarwal, C. C. (2015). *Data mining: the textbook*. Springer.

Ahmad, R., y Kamaruddin, S. (2012). An overview of time-based and condition-based maintenance in industrial application. *Computers & industrial engineering*, *63*(1), 135–149.

Aller, J. M. (2006). Máquinas eléctricas rotativas: Introducción a la teoría general. *Editorial Equinoccio*.

Alligood, K. T., Sauer, T. D., y Yorke, J. A. (1996). *Chaos*. Springer.

Al-Najjar, B., Ingwald, A., y Kans, M. (2016). Maintenance in real estate and manufacturing industries: Differences, problems, needs and potentials-four case studies. En *Proceedings of the 10th world congress on engineering asset management (wceam 2015)* (pp. 13–27).

Amador, A., Bueno, A., y Amador, J. M. (2009). Modelo dinámico de la máquina sincrónica de polos salientes en vectores espaciales y su aplicación al control directo de par. *Ingeniería Energética*, *30*(2), 26–35.

Babu, A. H., Reddy, B. A., Naresh, P., y Reddy, M. S. (2016, 02). Design and static analysis of automobile gearbox cover. *World Wide Journal of Multidisciplinary Research and Development*, *WWJMRD 2016; 2(1): 28-37*.

Barber, C. B., Dobkin, D. P., y Huhdanpaa, H. (1996). The quickhull algorithm for convex hulls. *ACM Transactions on Mathematical Software (TOMS)*, *22*(4), 469–483.

Bayar, N., Darmoul, S., Hajri-Gabouj, S., y Pierreval, H. (2015). Fault detection, diagnosis and recovery using artificial immune systems: A review. *Engineering Applications of Artificial Intelligence*, *46*, 43–57.

Bevilacqua, M., y Braglia, M. (2000). The analytic hierarchy process applied to maintenance strategy selection. *Reliability Engineering & System Safety*, *70*(1), 71–83.

Breiman, L. (1996). Bagging predictors. *Machine learning*, *24*(2), 123–140.

Breiman, L. (2001). Random forests. *Machine learning*, *45*(1), 5–32. doi: 10.1023/a:1010933404324

Breiman, L. (2006). Random forest user notes. *Statistics Department, University of California, Berkeley*.

Breiman, L., Friedman, J. H., Olshen, R. A., y Stone, C. J. (1984). Classification and regression trees. belmont, ca: Wadsworth. *International Group*, *432*, 151–166.

Cerrada, M., Macancela, J.-C., Cabrera, D., Estupiñan, E., Sánchez, R.-V., y Medina, R. (2020). Reciprocating compressor multi-fault classification using symbolic dynamics and complex correlation measure. *Applied Sciences*, *10*(7), 2512.

Chingal Imaicela, D. E. (2018). *Adquisición de señales de corriente del motor de inducción combinando fallos en la maquinaria rotativa y elaboración de una guía de práctica sobre detección de fallos por medio del afcm* (B.S. thesis). Universidad Politécnica Salesiana.

Coria, V., Maximov, S., Rivas-Dávalos, F., Melchor, C., y Guardado, J. L. (2015). Analytical method for optimization of maintenance policy based on available system failure data. *Reliability Engineering & System Safety*, *135*, 55–63.

Cover, T., y Hart, P. (1967). Nearest neighbor pattern classification. *IEEE transactions on information theory*, *13*(1), 21–27.

Dasarathy, B. V. (1991). Nearest neighbor (nn) norms: Nn pattern classification techniques. *IEEE Computer Society Tutorial*.

El Naqa, I., y Murphy, M. J. (2015). What is machine learning? *Machine Learning in Radiation Oncology*, 3–11. doi: 10.1007/978-3-319-18305-3_1

Feng, Z., Liang, M., y Chu, F. (2013). *Recent advances in time–frequency analysis methods for machinery fault diagnosis: A review with application examples. mechanical systems and signal processing.*

Franklin, J. (2005). The elements of statistical learning: data mining, inference and prediction. *The Mathematical Intelligencer*, *27*(2), 83–85.

García Ruiz de León, M. (2018). *Análisis de sensibilidad mediante random forest* (B.S. thesis, Universidad Politécnica de Madrid). Descargado de `https://oa.upm.es/53368/`

Guo, G., Wang, H., Bell, D., y Bi, Y. (2004, 08). Knn model-based approach in classification. *On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE*, 986–996. doi: 10.1007/978-3-540-39964-3_62

Guo, X., y Hao, P. (2021). Using a random forest model to predict the location of potential damage on asphalt pavement. *Applied Sciences*, *11*(21), 10396.

Halkidi, M., y Vazirgiannis, M. (2002). Clustering validity assessment using multi representatives. En *Proceedings of the hellenic conference on artificial intelligence, setn* (pp. 237–249).

Isermann, R. (2005). *Fault-diagnosis systems: an introduction from fault detection to fault tolerance. springer science and business media.* Springer.

Jardine, A. K., Lin, D., y Banjevic, D. (2006). A review on machinery diagnostics and prognostics implementing condition-based maintenance. *Mechanical systems and signal processing*, *20*(7), 1483–1510.

Kan, M. S., Tan, A. C., y Mathew, J. (2015). A review on prognostic techniques for non-stationary and non-linear rotating systems. *Mechanical Systems and Signal Processing*, *62*, 1–20.

Kia, S. H., Henao, H., y Capolino, G.-A. (2009). Torsional vibration effects on induction machine current and torque signatures in gearbox-based electromechanical system. *IEEE Transactions on Industrial Electronics*, *56*(11), 4689–4699.

Kononenko, I. (1994). Estimating attributes: analysis and extensions of relief. in european conference on machine learning. *Machine Learning: ECML-94*, 172181.

Liu, H., Motoda, H., Setiono, R., y Zhao, Z. (2010). Feature selection: An ever evolving frontier in data mining. En *Feature selection in data mining* (pp. 4–13).

Liu, R., Yang, B., Zio, E., y Chen, X. (2018). Artificial intelligence for fault diagnosis of rotating machinery: A review. *Mechanical Systems and Signal Processing*, *108*, 33–47.

Llivicura Orellana, H. F. (2019). *Señales de vibración: evaluación de indicadores de condición extraídos del dominio de frecuencia para el diagnóstico de fallos en cajas de engranajes rectos* (B.S. thesis). Universidad Politécnica Salesiana.

Loaiza Sánchez, W. F. (2021). *Detección y diagnóstico de fallos de caja de engranajes rectos utilizando un algoritmo de clasificación basado en similaridad difusa aplicado en señales de vibración* (B.S. thesis). Universidad Politécnica Salesiana.

Medina, R., Alvarez, X., Jadán, D., Cerrada, M., Sánchez, R.-V., y Macancela, J. C. (2017). Poincaré plot features from vibration signal for gearbox fault diagnosis. En *2017 ieee second ecuador technical chapters meeting (etcm)* (pp. 1–6).

Mejía, M. V. (2022). *Análisis exploratorio de datos para la selección de atributos de poincaré aplicados a la señal de par eléctrico de un motor de inducción bajo perturbaciones de carga por fallos en una caja de engranajes* [Report]. (unpublished)

Mitchell, M. W. (2011). Bias of the random forest out-of-bag (oob) error for certain input parameters. *Open Journal of Statistics*, *1*(03), 205.

Ortega Lucero, L. R. (2021). *Estimación del par de carga en motores de inducción basado en su modelo matemático y orientado a la detección de fallos en cajas de engranajes* (B.S. thesis). Universidad Politécnica Salesiana.

Peña, M., Cerrada, M., Alvarez, X., Jadán, D., Lucero, P., Milton, B., ... Sánchez, R.-V. (2018). Feature engineering based on anova, cluster validity assessment and knn for fault diagnosis in bearings. *Journal of Intelligent & Fuzzy Systems*, *34*(6), 3451–3462.

Peña, M., Cerrada, M., Medina, R., Cabrera, D., y Sánchez, R. V. (2022). Poincaré plot features and statistical features from current and vibration signals for fault severity classification of helical gear tooth breaks. *Journal of Computing and Information Science in Engineering*, *23*(2), 021009.

Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia*, *4*(2), 1883.

R, K., S, H., H, y W, V., H. (2006). *System health monitoring and prognostics a review of*

current paradigms and practices. Springer.

Rengifo, J., Aller, J. M., Bueno, A., Viola, J., y Restrepo, J. (2012). Parameter estimation method for induction machines using the instantaneous impedance during a dynamic start-up. En *2012 vi andean region international conference* (pp. 11–14).

Robnik-Šikonja, M., y Kononenko, I. (2003). Theoretical and empirical analysis of relieff and rrelieff. *Machine learning*, *53*(1), 23–69.

Rodríguez, J., Rodríguez, A., Dos Santos, M., Peña, C., Botelho, R., Cunha, F., y Pérez, O. (2014). Deterioro y modos de fallo en engranajes. En *presentado en viii conferencia científica internacional de ingeniería mecánica.* doi: 10.13140/RG.2.1.4579.3766

Samanthula, B. K., Elmehdwi, Y., y Jiang, W. (2014). K-nearest neighbor classification over semantically secure encrypted relational data. *IEEE transactions on Knowledge and data engineering*, *27*(5), 1261–1273.

Sanchez, R. V., Lucero, P., Macancela, J. C., Cerrada, M., Cabrera, D., y Vasquez, R. (2019). Gear crack level classification by using knn and time-domain features from acoustic emission signals under different motor speeds and loads. En *Proceedings-2018 international conference on sensing, diagnostics, prognostics, and control, sdpc 2018* (Vol. 11).

Sánchez Loja, R. V. (2018). *Diagnóstico de fallos en cajas de engranajes con base en la fusión de datos de señales de vibración, corriente y emisión acústica* (Ph.D thesis). Universidad Pontificia Bolivariana.

Scarf, P. (2007). A framework for condition monitoring and condition based maintenance. *Quality Technology & Quantitative Management*, *4*(2), 301–312.

Schwarz, D. F., König, I. R., y Ziegler, A. (2010). On safari to random jungle: a fast implementation of random forests for high-dimensional data. *Bioinformatics*, *26*(14), 1752–1758.

Sharma, N., y Saroha, K. (2015). Study of dimension reduction methodologies in data mining. En *International conference on computing, communication & automation* (pp. 133–137).

Winn, J., Bishop, C., y Diethe, T. (2015). Model-based machine learning. En (p. 26–33). Chapman & amp; Hall/CRC.

Wu, S., y Chow, T. W. (2004). Clustering of the self-organizing map using a clustering validity index based on inter-cluster and intra-cluster density. *Pattern Recognition*, *37*(2), 175–188.

Wu, X., Kumar, V., Quinlan, J. R., Ghosh, J., Yang, Q., Motoda, H., ... others (2007). Top 10 algorithms in data mining. *Knowledge and information systems*, *14*(1), 1–37. doi:

10.1007/s10115-007-0114-2

Yang, B.-S., Di, X., y Han, T. (2008). Random forests classifier for machine fault diagnosis. *Journal of mechanical science and technology*, *22*(9).

Yang, B. S., Park, C. H., y Kim, H. J. (2000). An efficient method of vibration diagnostics for rotating machinery using a decision tree. *International Journal of Rotating Machinery*, *6*(1), 19–27.

Ziegler, A., y König, I. R. (2014). Mining data with random forests: current options for real-world applications. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *4*(1), 55–63.

ANEXOS

Anexo A: Matriz de Consistencia Lógica

Tabla 15

Matriz de consistencia lógica.

Detección de fallos en una caja de engranes usando algoritmos de aprendizaje automático y mapas de Poincaré aplicados a la señal de par eléctrico				
PROBLEMA GENERAL	**OBJETIVO GENERAL**	**HIPÓTESIS GENERAL**	**VARIABLES**	**MARCO TEÓRICO**
¿Es posible mediante un modelo de aprendizaje automático detectar diferentes niveles de severidad de fallo de diente roto en una caja de engranes rectos usando la señal del par eléctrico?	Diagnosticar diferentes niveles de severidad de diente roto en una caja de engranes a partir del análisis de la señal del par eléctrico usando algoritmos de aprendizaje automático supervisados.	El par eléctrico es una señal sensible a fallos en una caja de engranes y puede usarse en modelos de detección de fallos ajustados con algoritmos de aprendizaje automático.	**VI:**Atributos del par eléctrico **VD:** Precisión de clasificación de los modelos	Par eléctrico, mantenimiento basado en la condición,caja de engranes, modo de falla de diente roto y plan experimental.
ESPECÍFICOS	**ESPECÍFICOS**	**ESPECÍFICAS**		
¿Cuál es el algoritmo de aprendizaje automático en un entorno supervisado que tiene mejor desempeño para la clasificación de niveles de severidad de falla de diente roto	Aplicar al menos dos algoritmos de aprendizaje automático para determinar el nivel de severidad de fallo de diente roto en una caja de engranajes a partir de atributos de Poincaré.	Al usar diferentes algoritmos de aprendizaje automático se obtienen diferentes modelos (KNN y Random Forest), lo cual permite conocer cuál de ellos tendrá un mejor desempeño al detectar niveles de severidad de fallo de diente roto de un engranaje.	**VI:** Algoritmo de aprendizaje **VD:** Modelo computacional	Diagramas de Poincarè y atributos, Chi Cuadrado, CDwb y ReliefF .
¿Es posible diagnosticar con precisión apropiada el grado de severidad en la rotura del diente en una caja de engranes a partir de la señal del par eléctrico usando un modelo computacional ajustado por un algoritmo de aprendizaje automático?	Desarrollar un estudio comparativo entre los diferentes modelos computacionales de aprendizaje automático obtenidos, con base a la precisión en clasificación.	Dependiendo tanto de los atributos usados como de los modelos computacionales, será posible realizar una comparación y determinar cuál de los mismos es el más preciso.	**VI:** Atributos obtenidos **VD:** Precisión en clasificación	Clasificación de fallos basados en datos y Machine Learning
¿Cuáles son los atributos más representativos a extraer de las señales del par eléctrico usando diagramas de Poincaré para la detección de fallos y usarlos en el algoritmo de aprendizaje automático?	Determinar la influencia de atributos de Poincaré extraídos del par eléctrico, en el desempeño de los modelos de aprendizaje automático con base a la precisión en clasificación.	Al usar los atributos extraídos de gráficos de Poincaré y seleccionados por técnicas, tales como: RefliefF, Chi Caudrado, y de CDbW es posible establecer los atributos adecuados para la detección de fallos usando diferentes modelos computacionales.	**VI:**Modelos computacionales **VD:** Precisión en clasificación	Algoritmos KNN y Randon Forest

Fuente: *Autor*

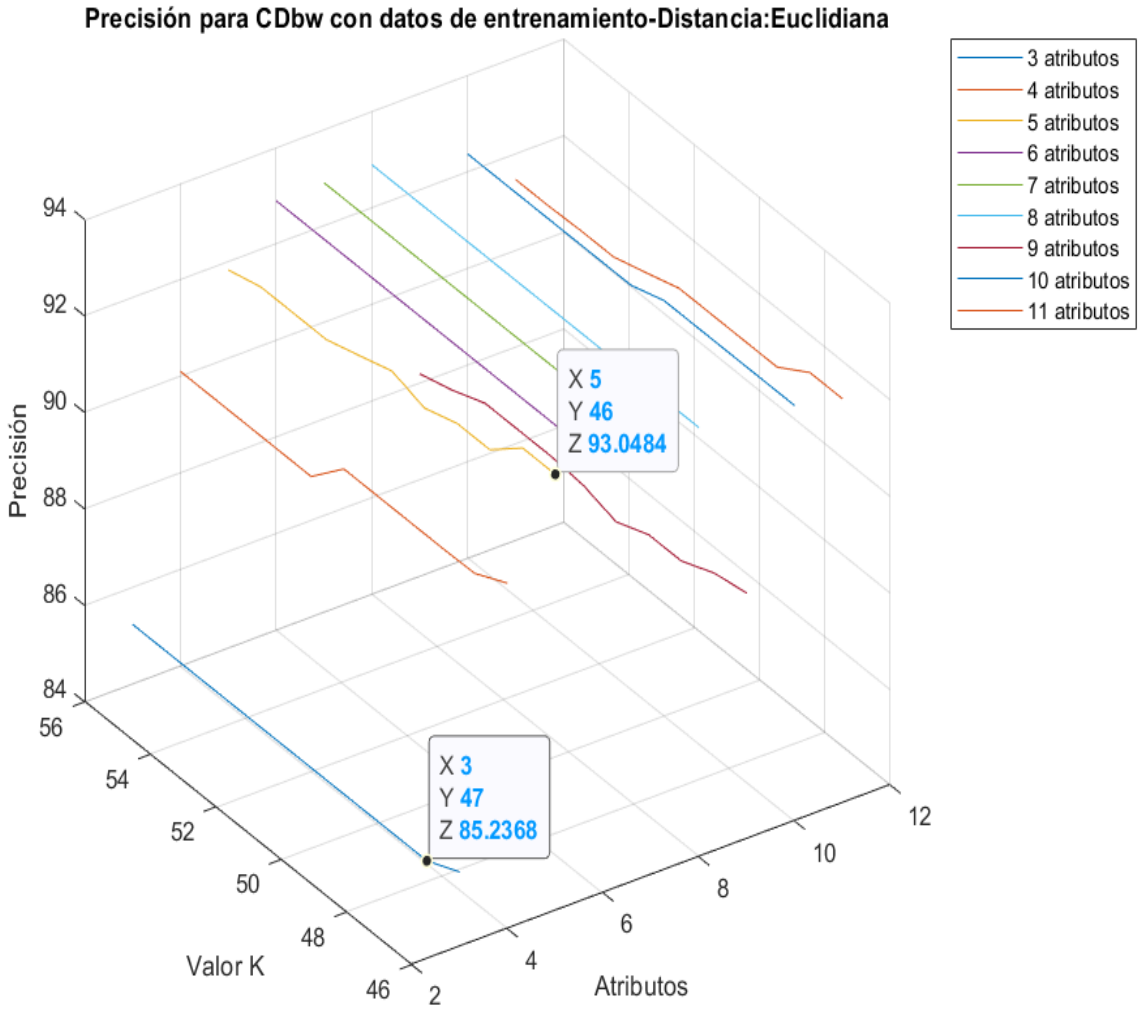
Anexo B: Gráficas de resultados

9.1. Resultados con KNN con datos de entrenamiento

9.1.1. Resultados con distancia Euclidiana

Figura 37

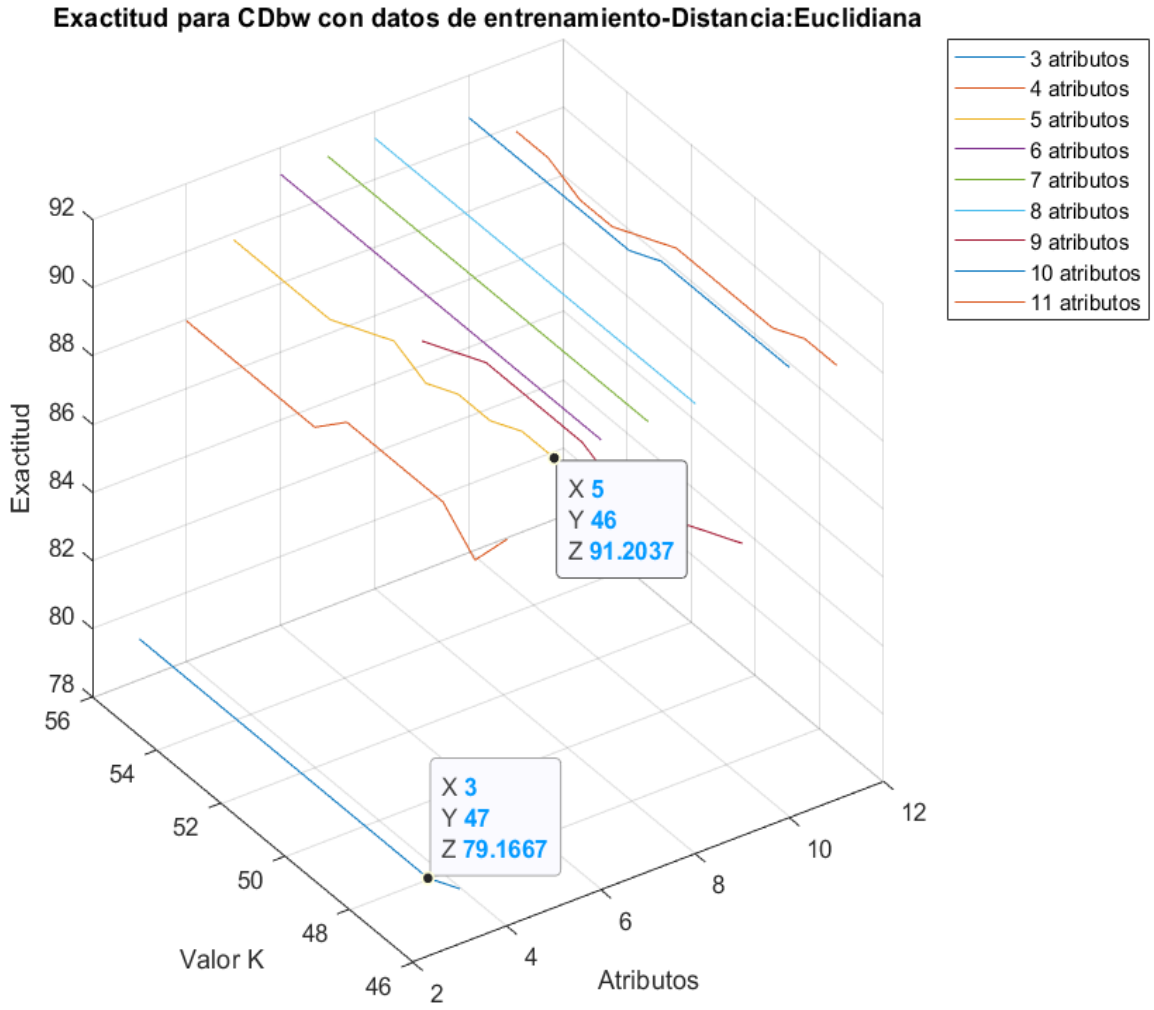
Precisión de CDbw con datos de entrenamiento con distancia Euclidiana.



Nota: Obtenido de Autor.

Figura 38

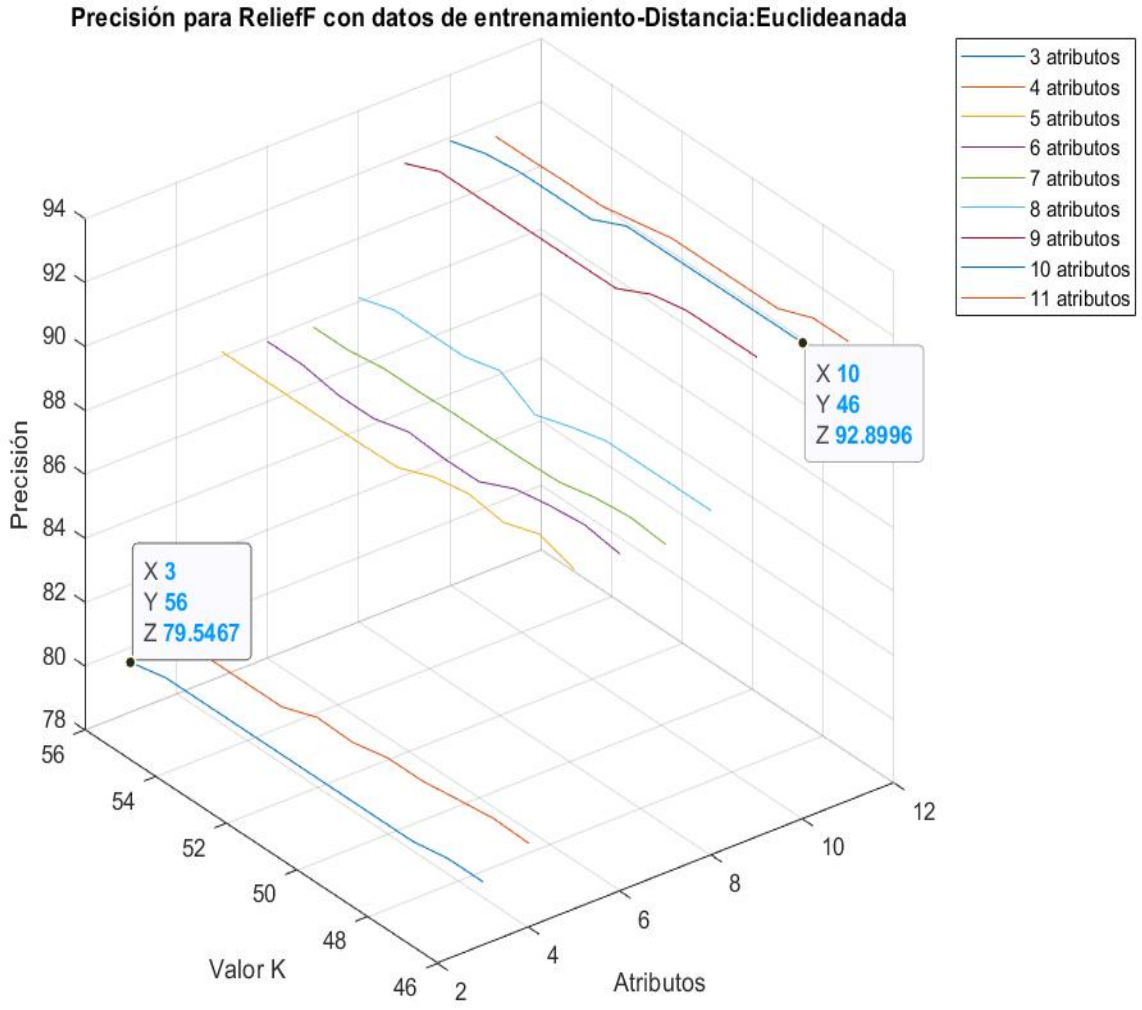
Exactitud de CDbw con datos de entrenamiento con distancia Euclidiana.



Nota: Obtenido de Autor.

Figura 39

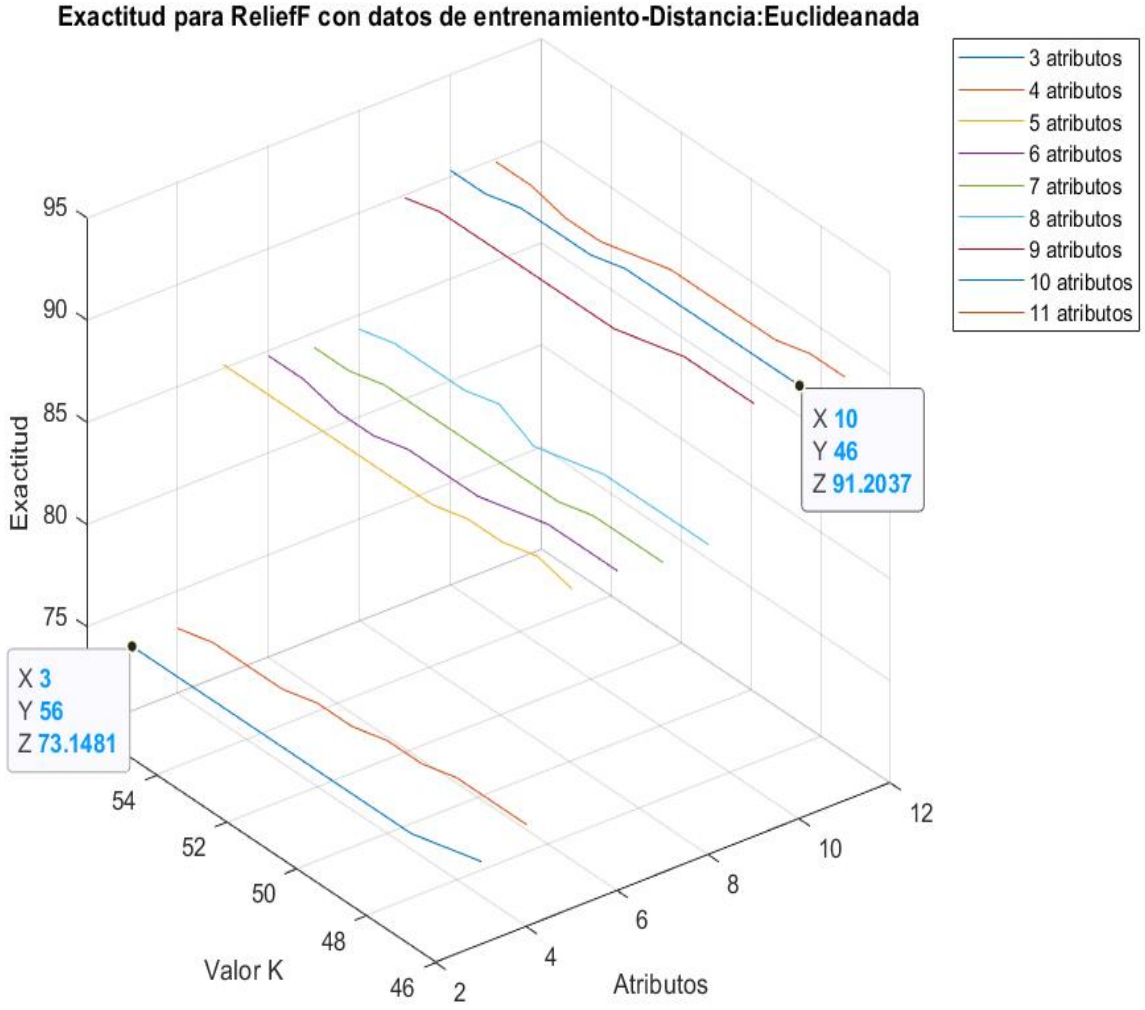
Precisión de ReliefF con datos de entrenamiento con distancia Euclidiana.



Nota: Obtenido de Autor.

Figura 40

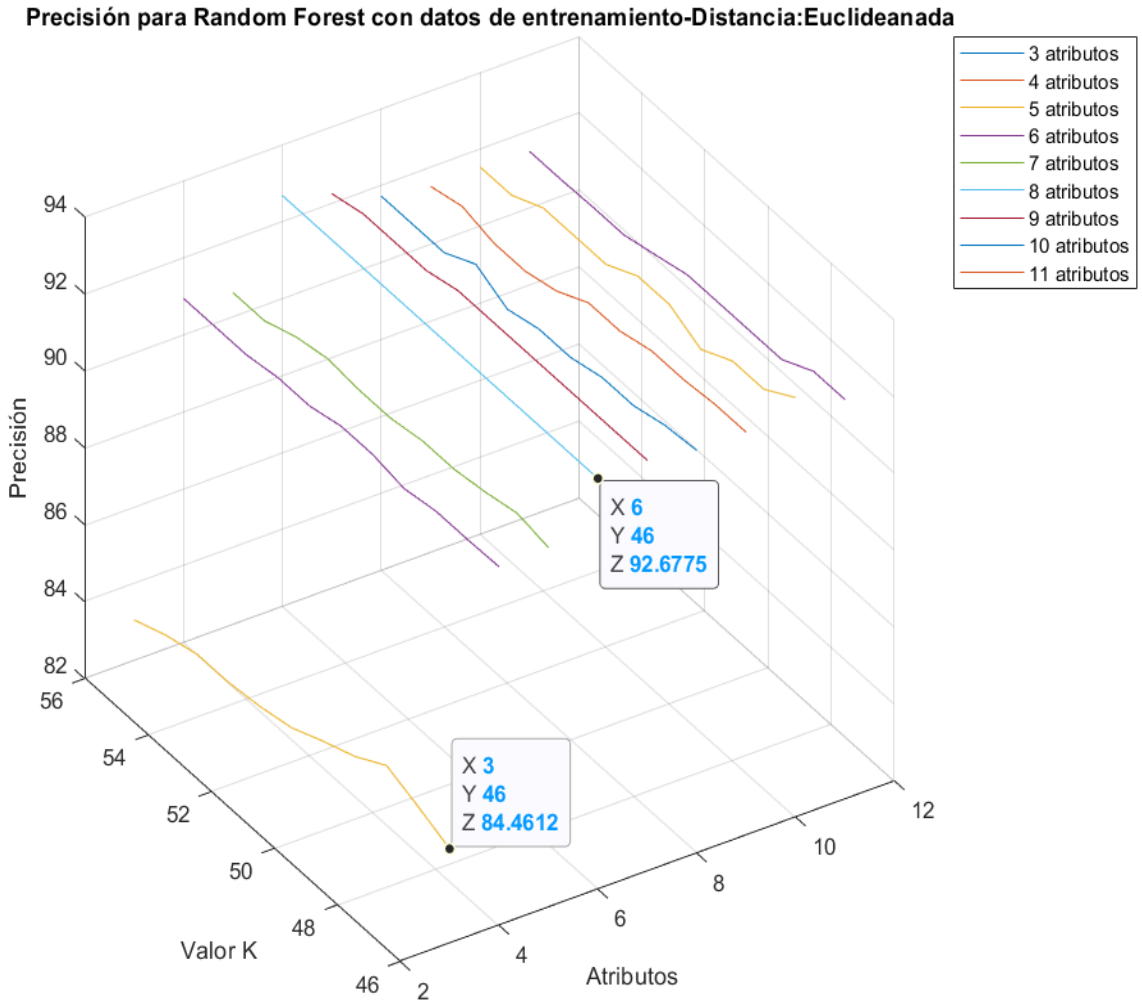
Exactitud de ReliefF con datos de entrenamiento con distancia Euclidiana.



Nota: Obtenido de Autor.

Figura 41

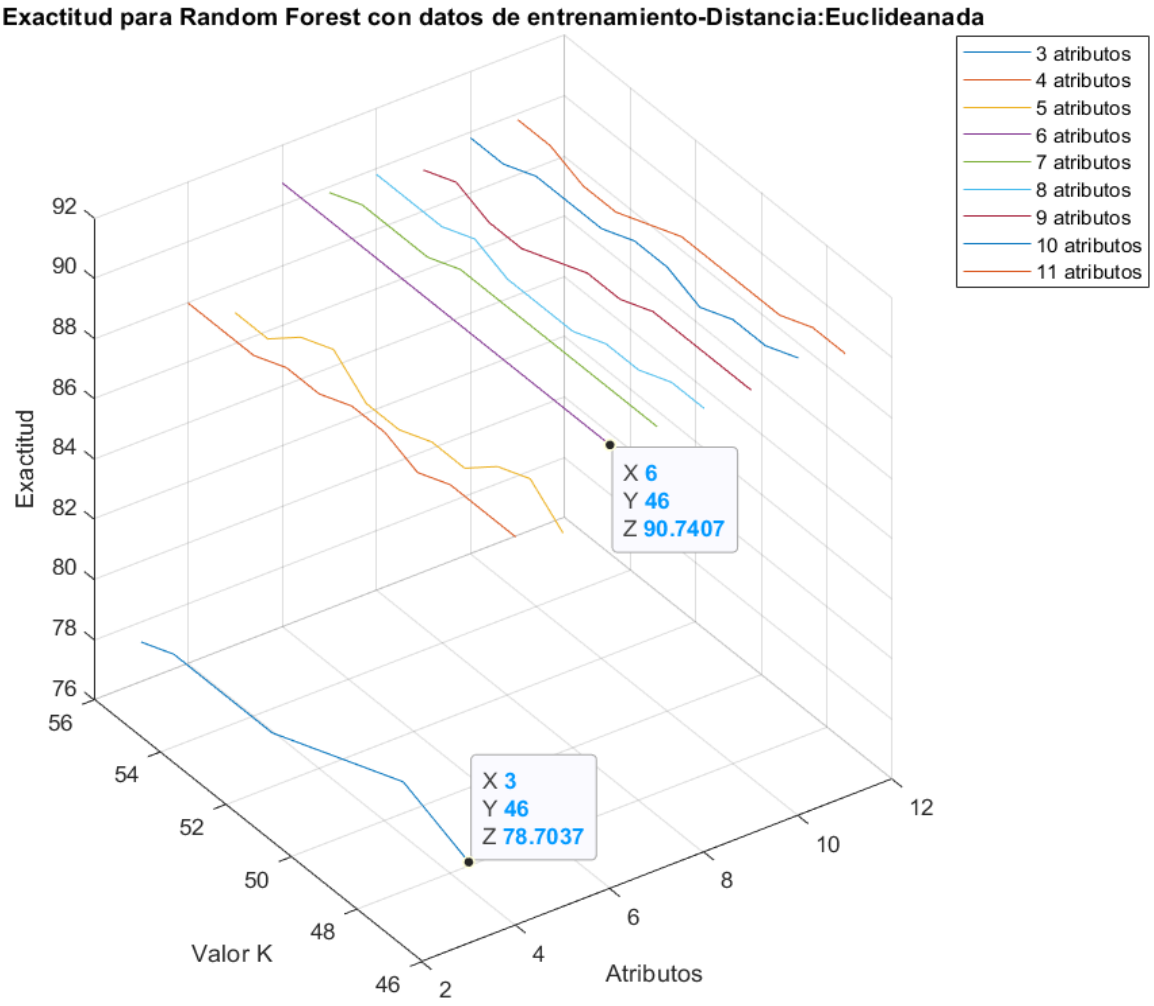
Precisión de Random Forest con datos de entrenamiento con distancia Euclidiana.



Nota: Obtenido de Autor.

Figura 42

Exactitud de Random Forest con datos de entrenamiento con distancia Euclidiana.

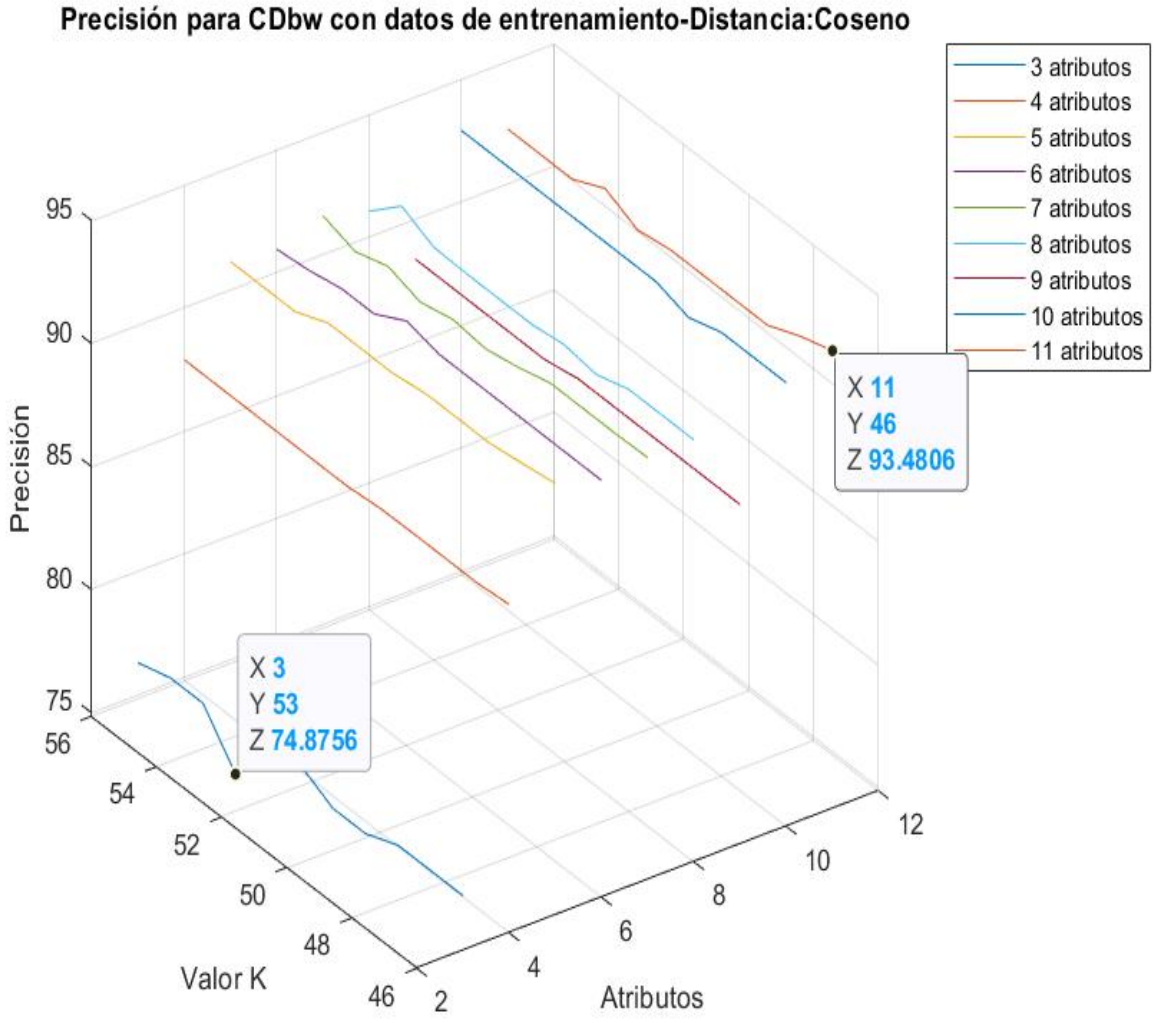


Nota: Obtenido de Autor.

9.1.2. Resultados con distancia Coseno

Figura 43

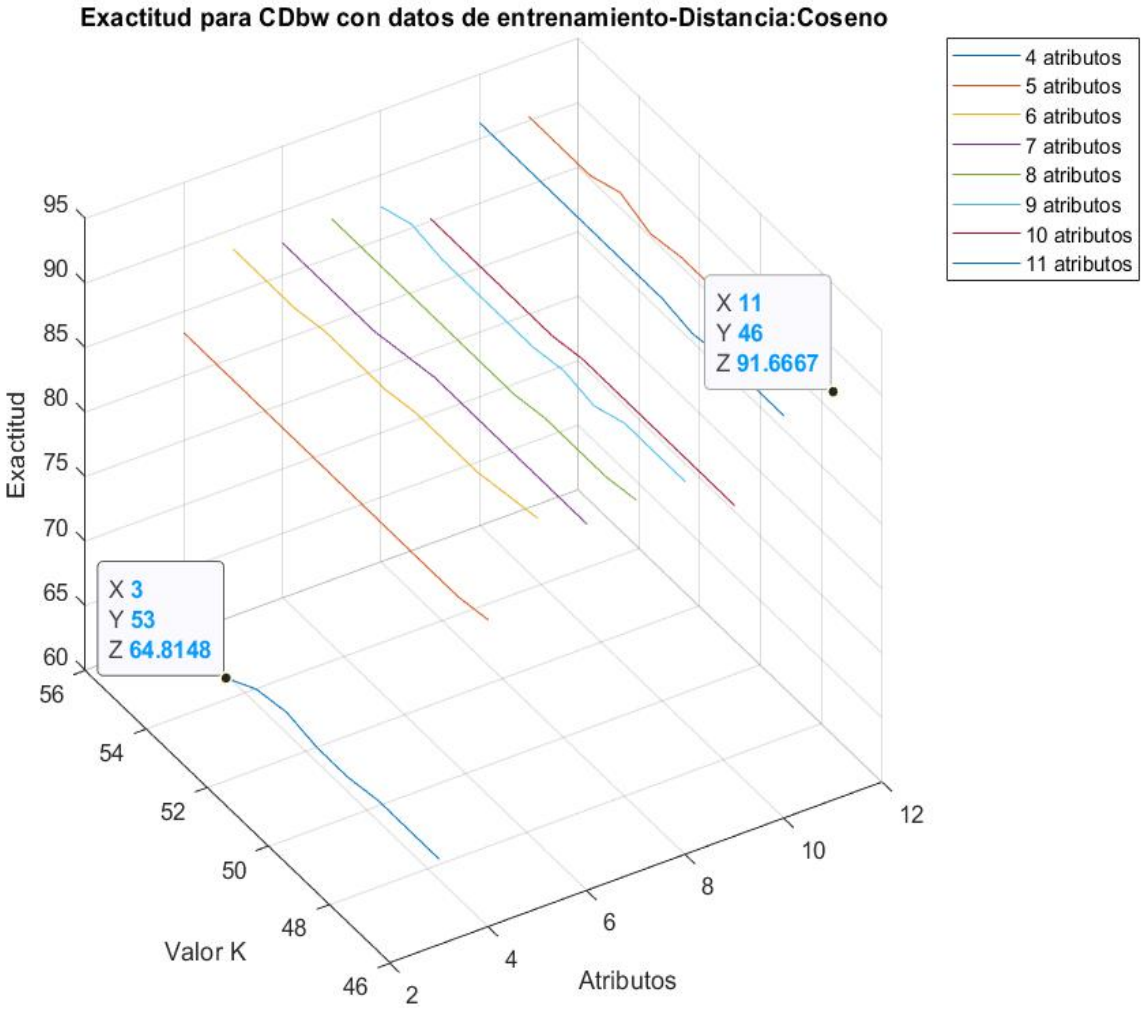
Precisión de CDbw con datos de entrenamiento con distancia Coseno.



Nota: Obtenido de Autor.

Figura 44

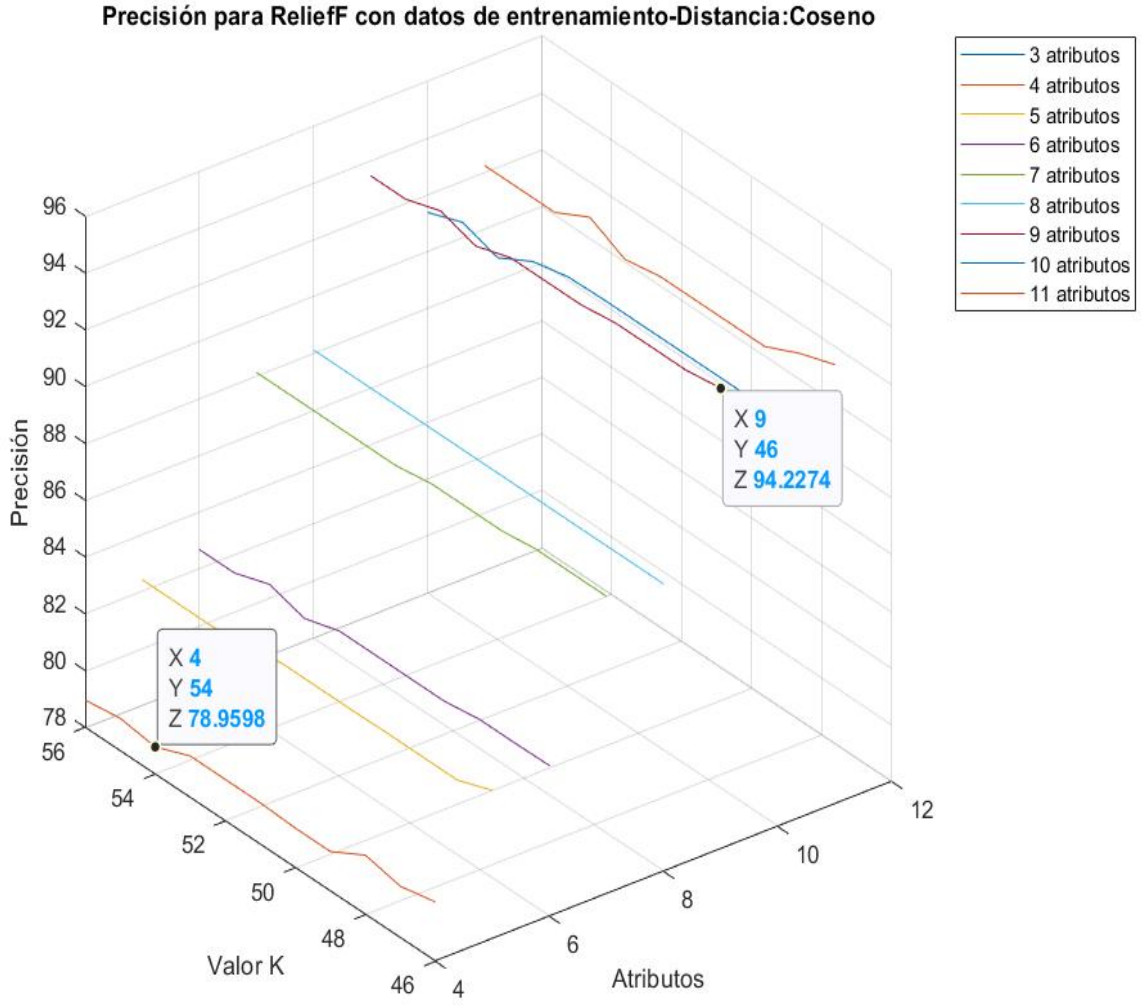
Exactitud de CDbw con datos de entrenamiento con distancia Coseno.



Nota: Obtenido de Autor.

Figura 45

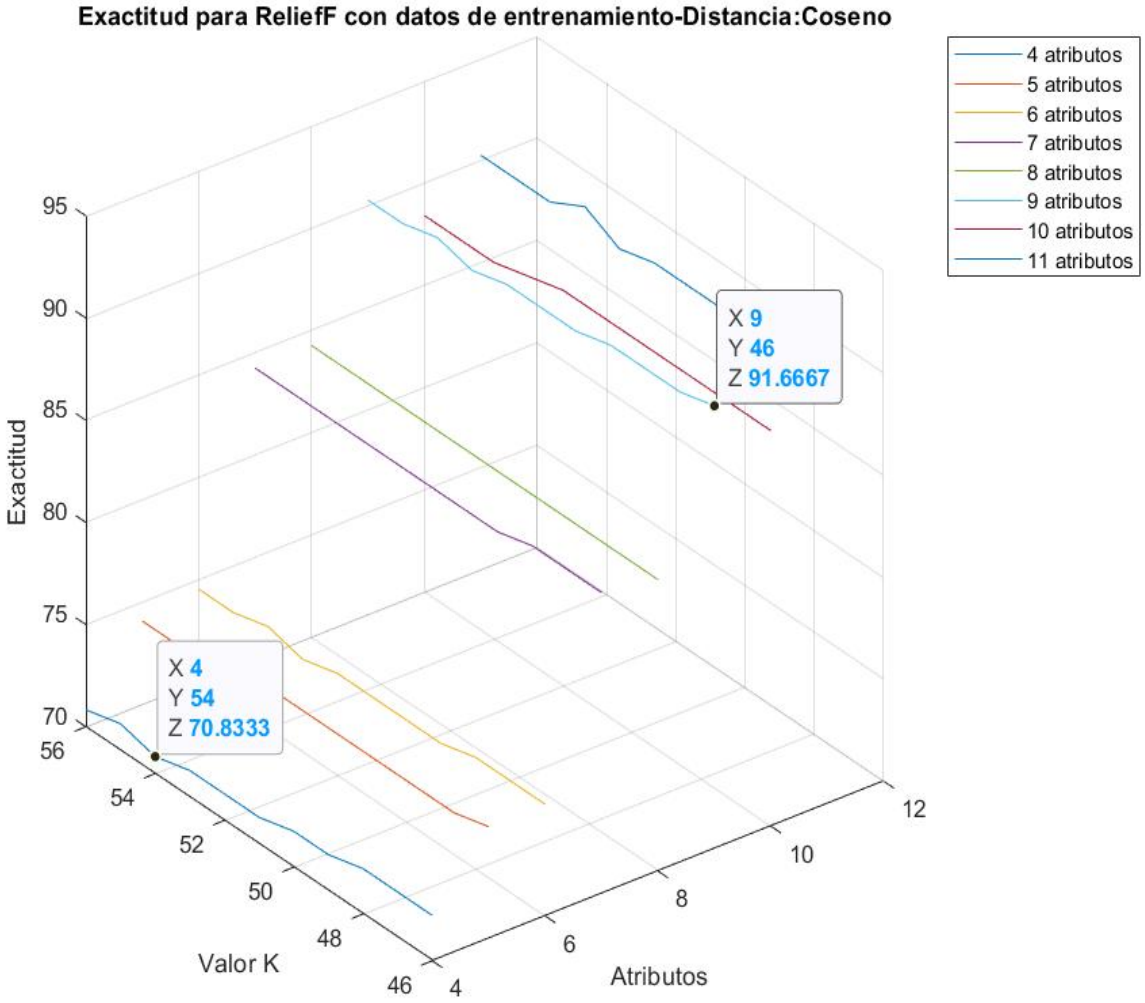
Precisión de ReliefF con datos de entrenamiento con distancia Coseno.



Nota: Obtenido de Autor.

Figura 46

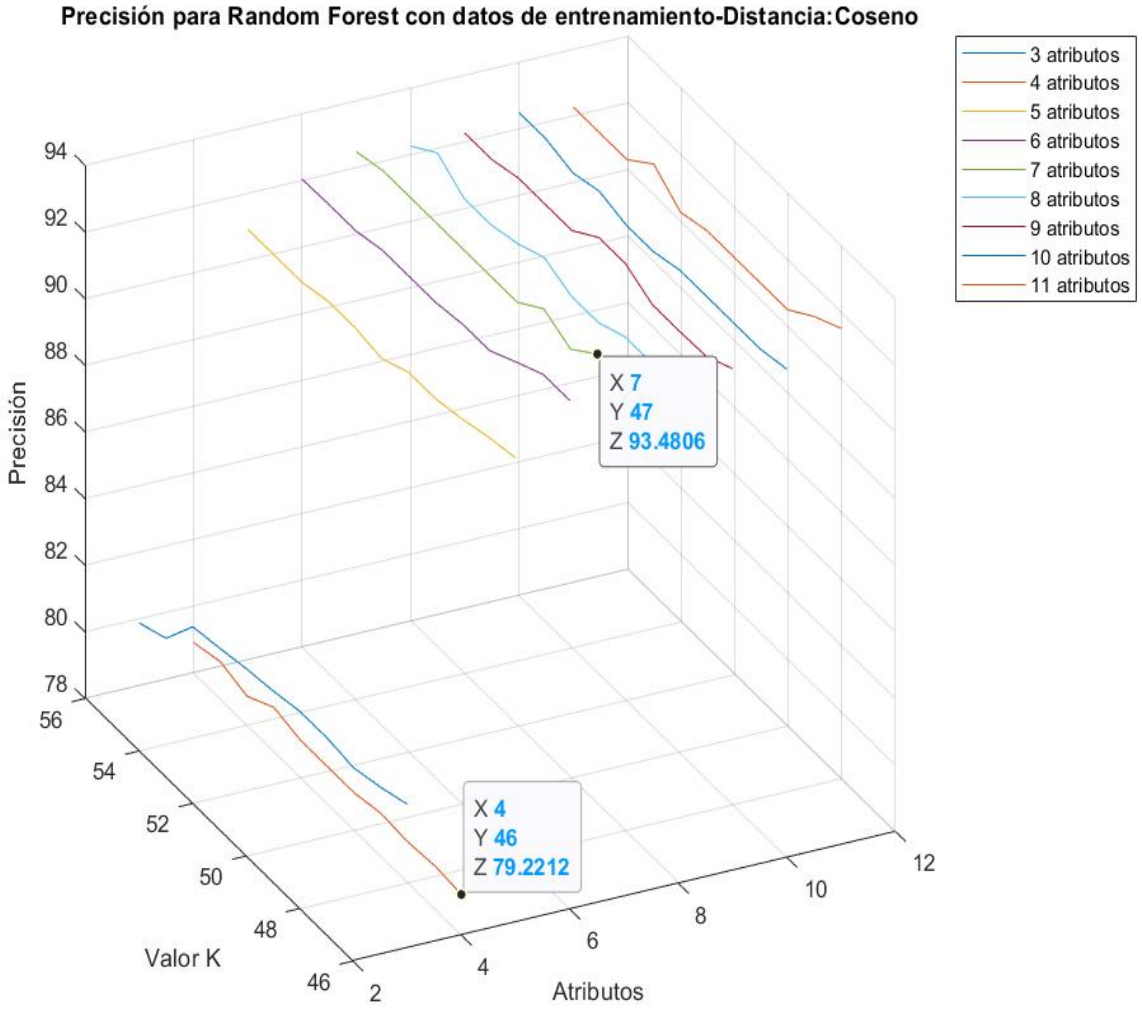
Exactitud de ReliefF con datos de entrenamiento con distancia Coseno.



Nota: Obtenido de Autor.

Figura 47

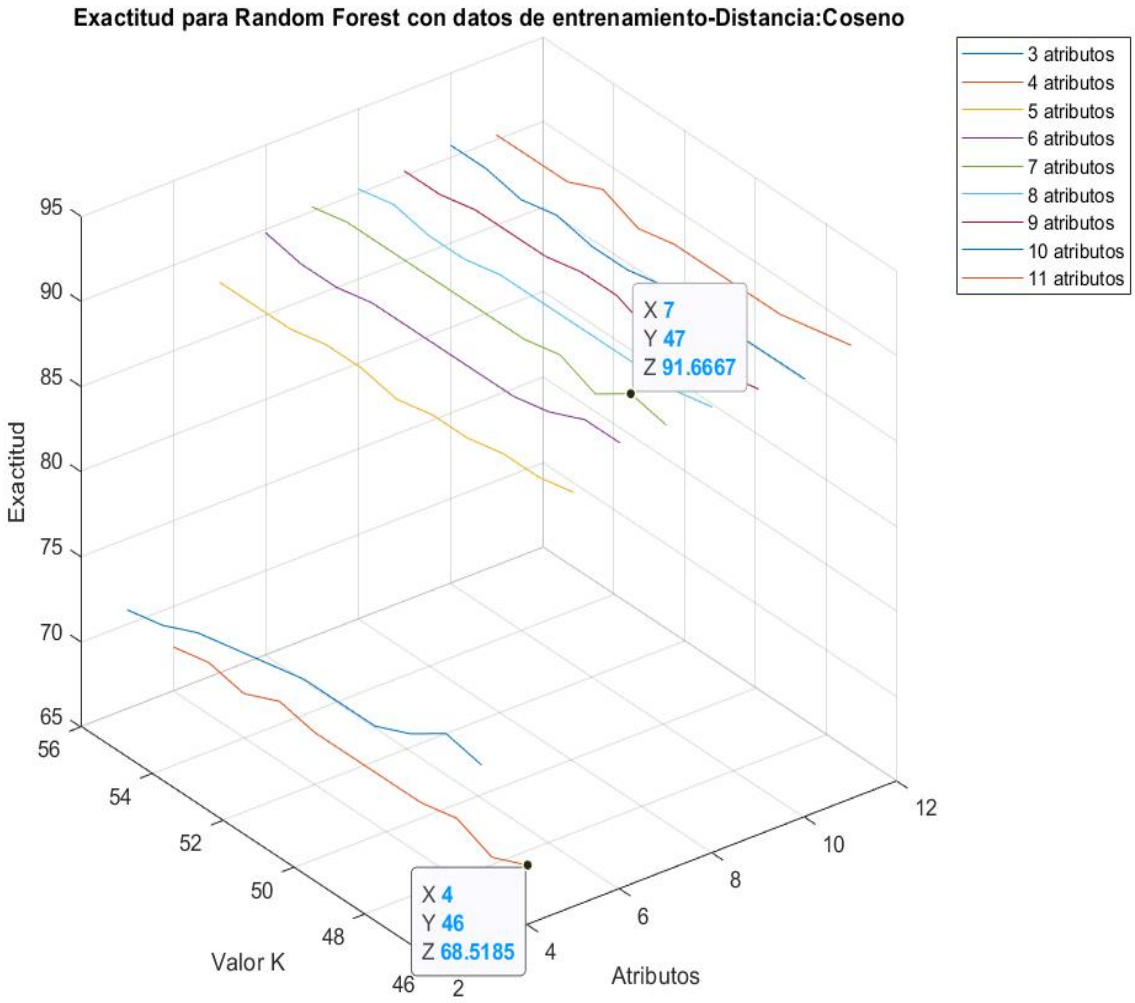
Precisión de ReliefF con datos de entrenamiento con distancia Coseno.



Nota: Obtenido de Autor.

Figura 48

Exactitud de Random Forest con datos de entrenamiento con distancia Coseno.

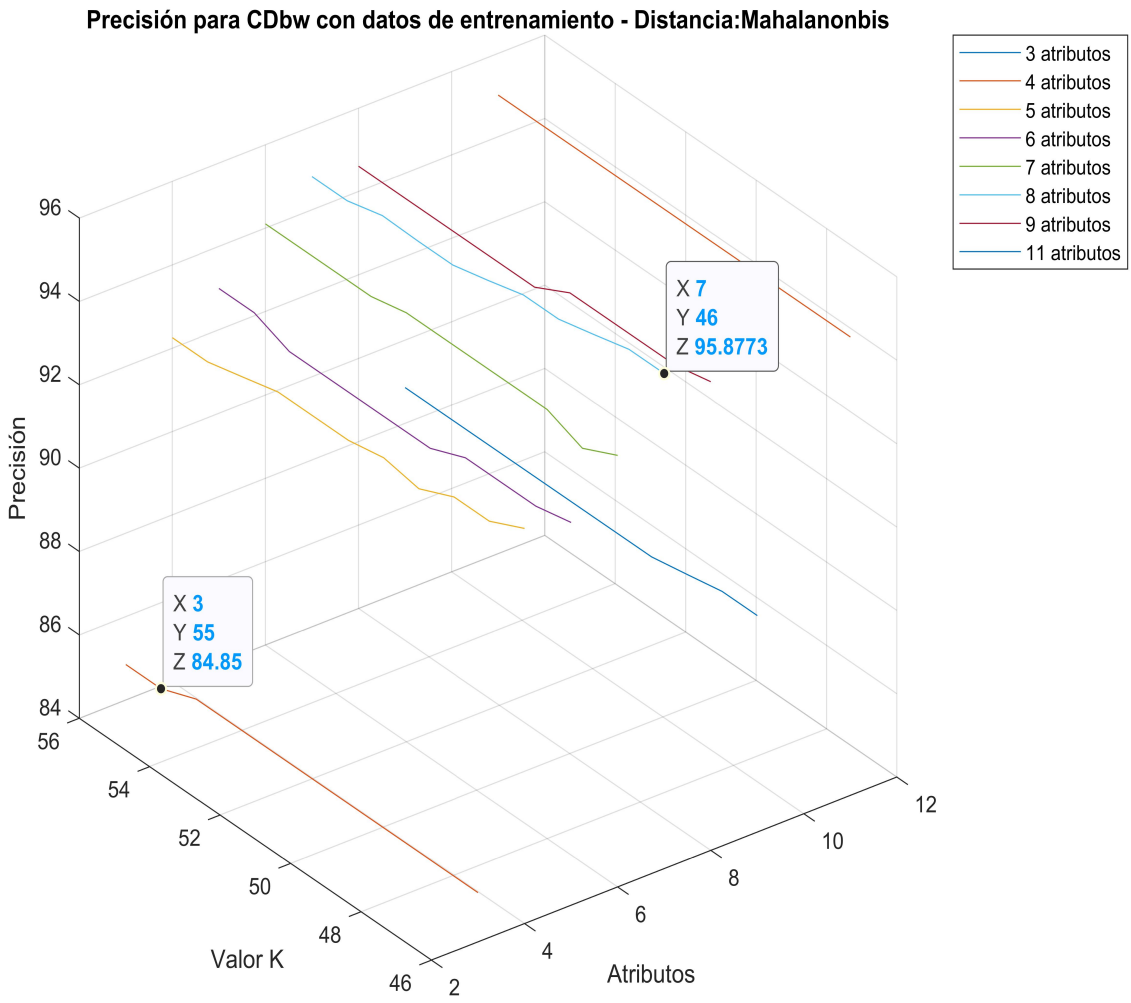


Nota: Obtenido de Autor.

9.1.3. Resultados con distancia Mahalanobis

Figura 49

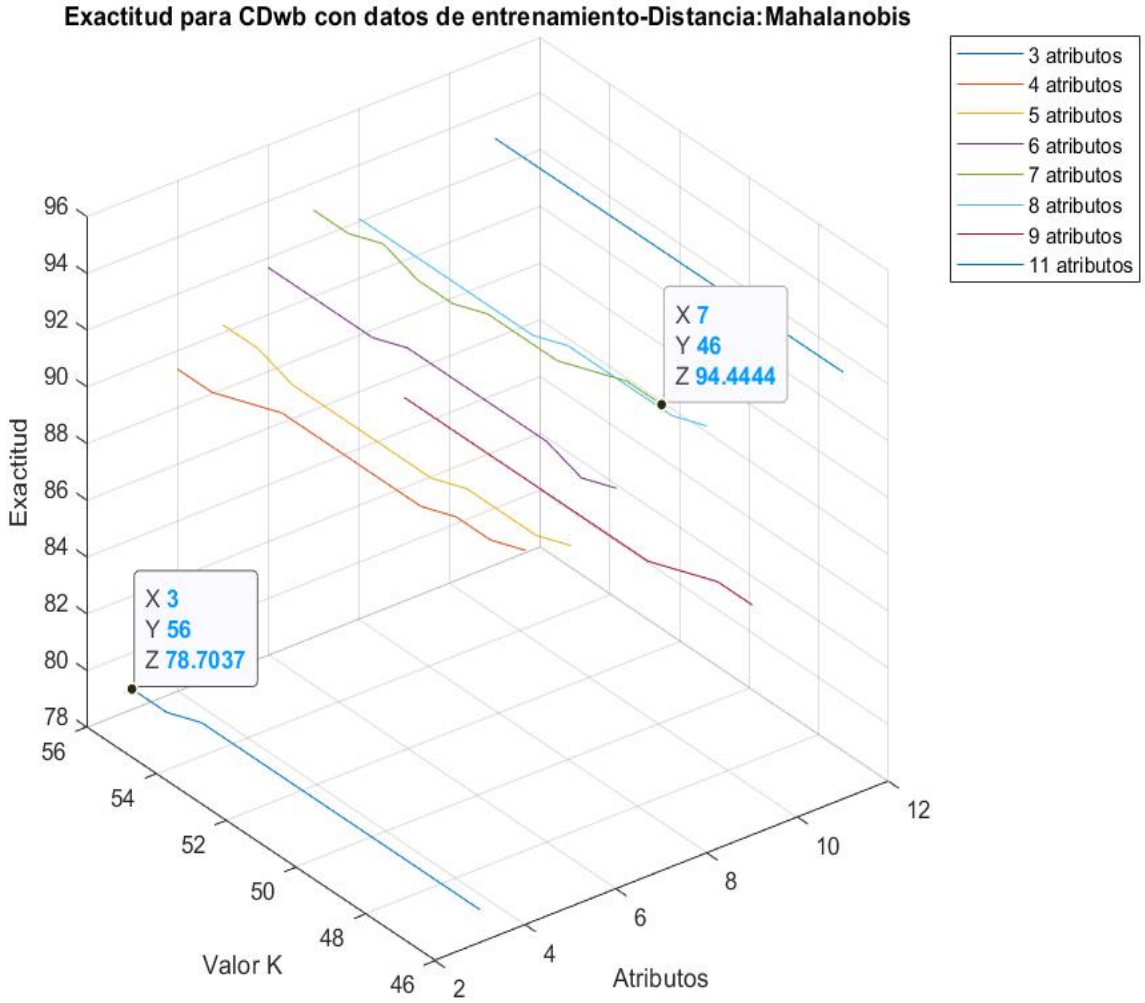
Precisión de CDbw con datos de entrenamiento con distancia Mahalanobis.



Nota: Obtenido de Autor.

Figura 50

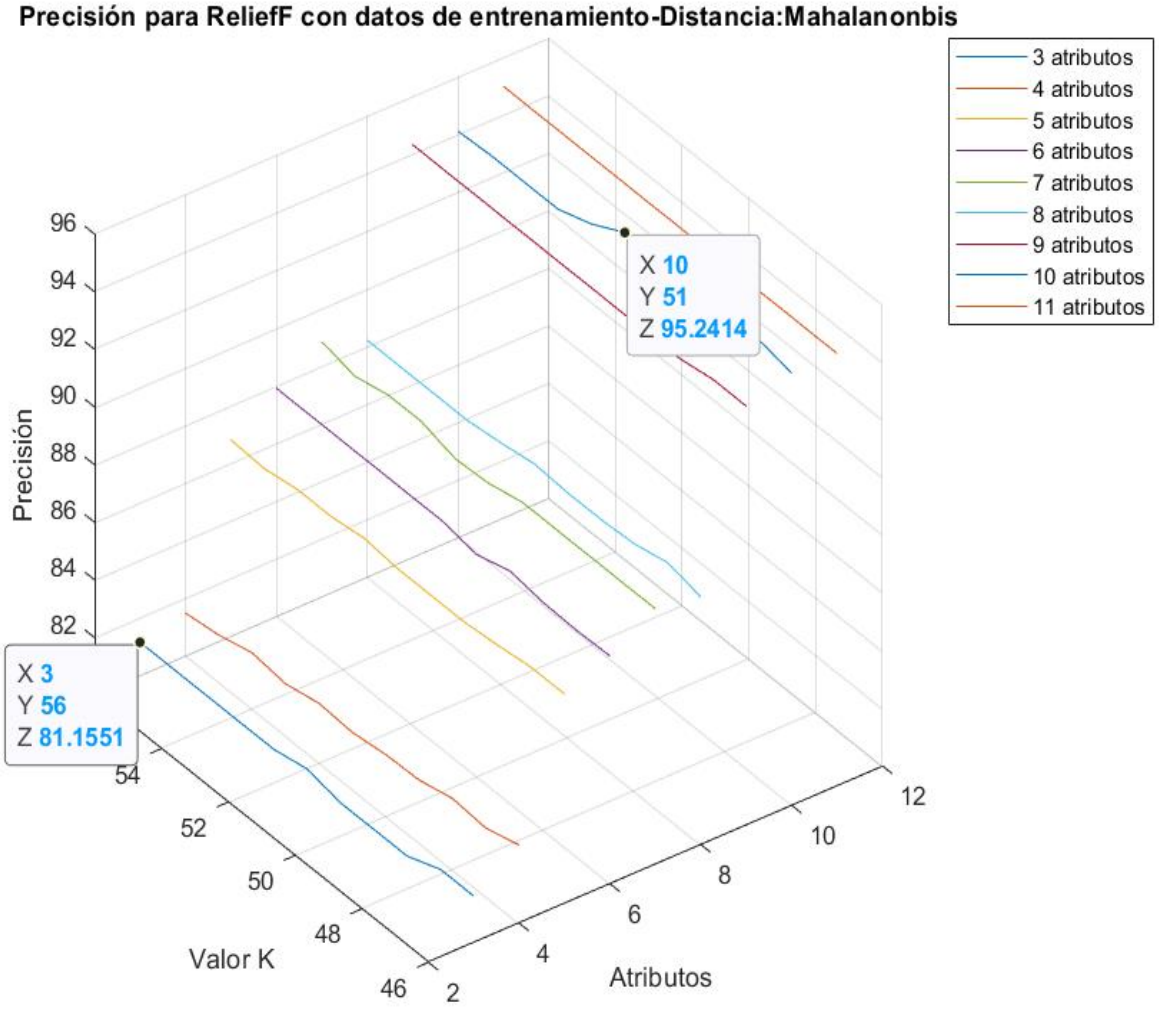
Exactitud de CDbw con datos de entrenamiento con distancia Mahalanobis.



Nota: Obtenido de Autor.

Figura 51

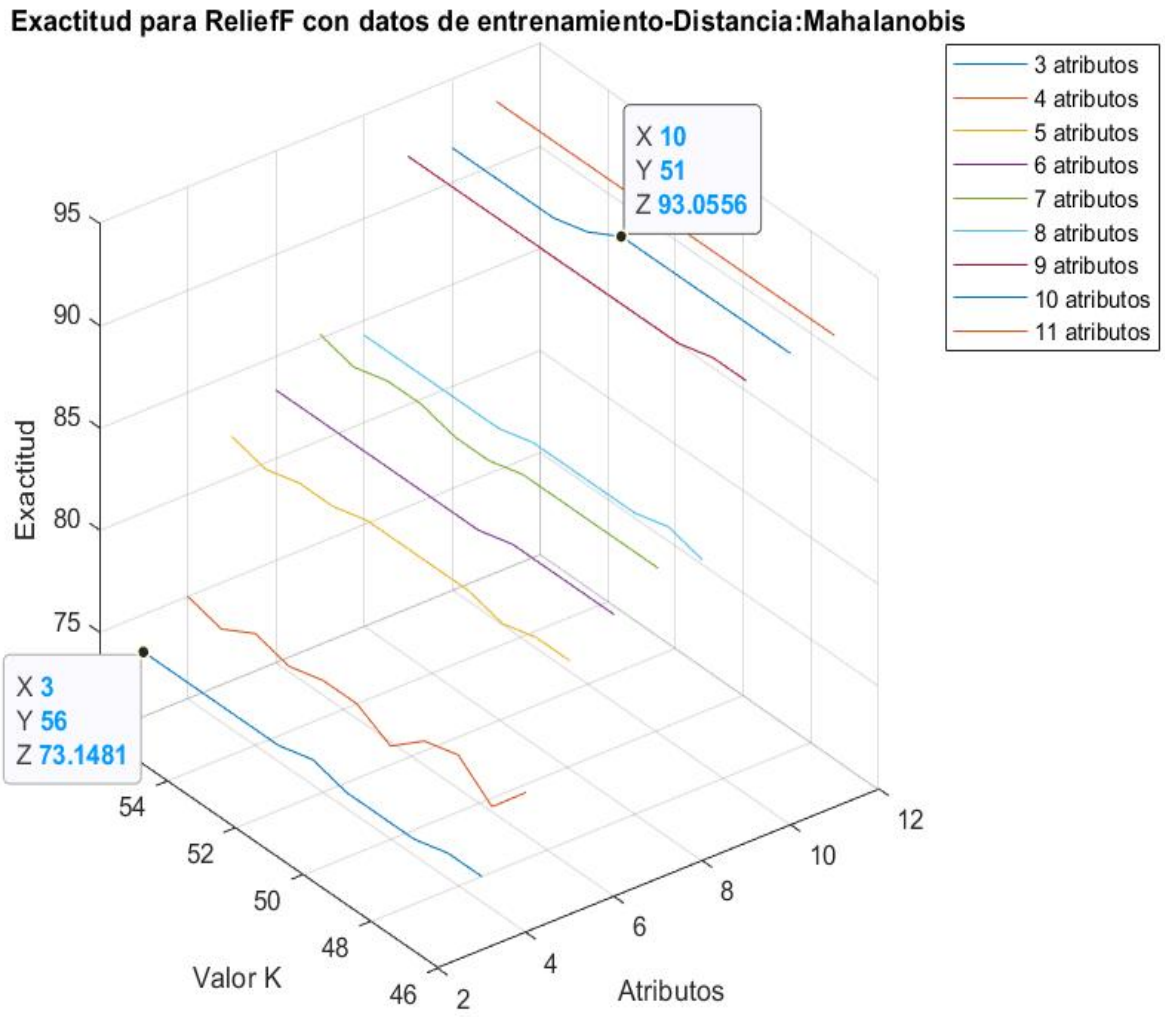
Precisión de ReliefF con datos de entrenamiento con distancia Mahalanobis.



Nota: Obtenido de Autor.

Figura 52

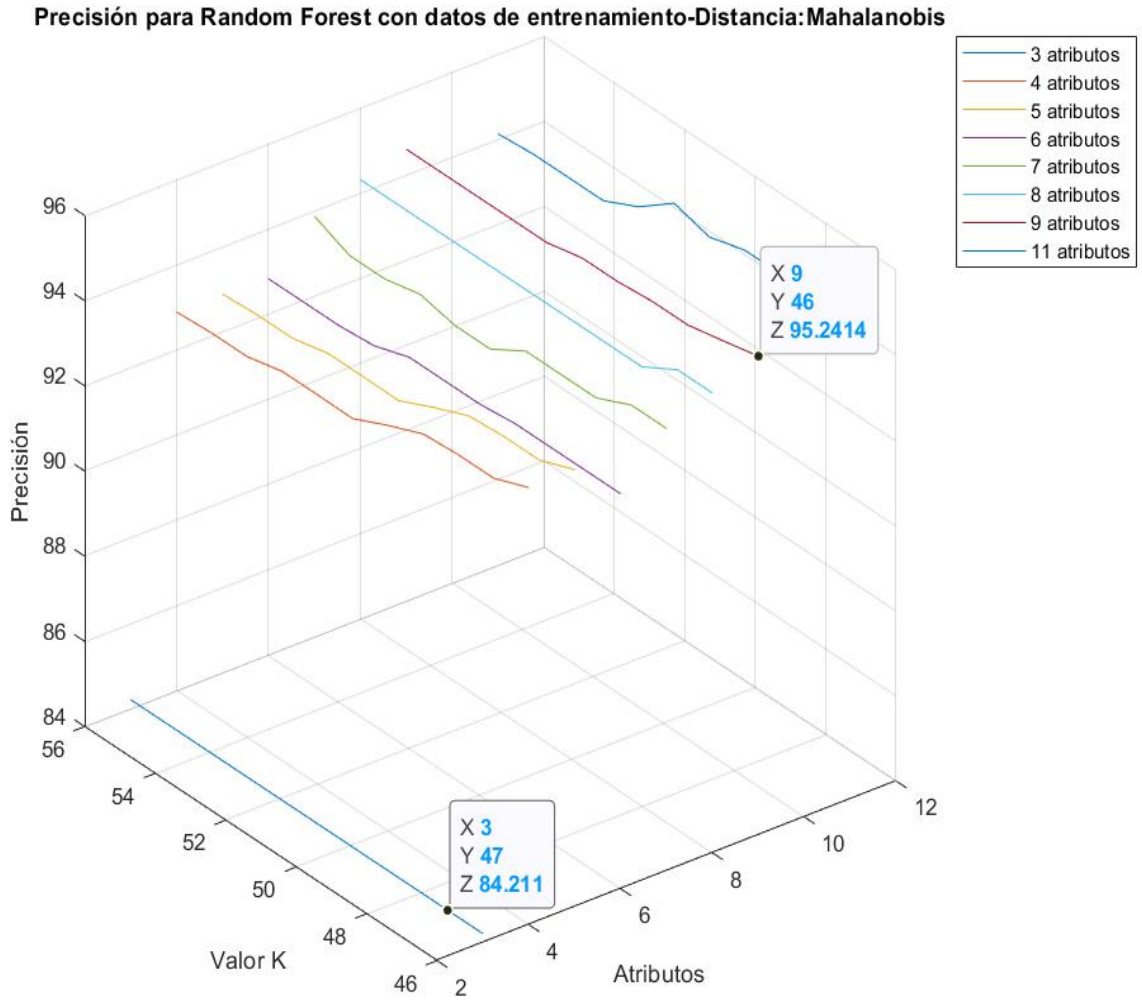
Exactitud de ReliefF con datos de entrenamiento con distancia Mahalanobis.



Nota: Obtenido de Autor.

Figura 53

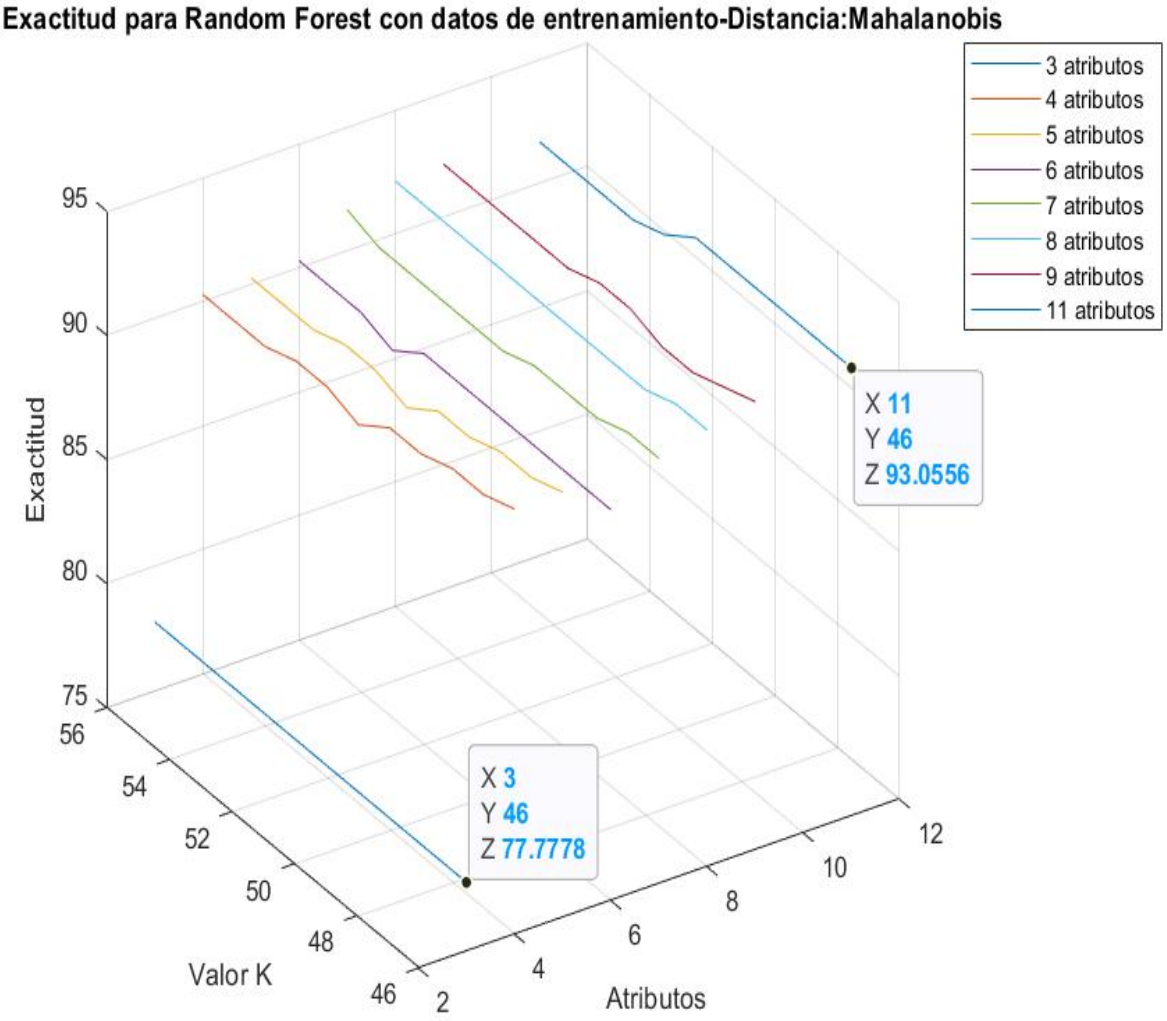
Precisión de Random Forest con datos de entrenamiento con distancia Mahalanobis.



Nota: Obtenido de Autor.

Figura 54

Exactitud de Random Forest con datos de entrenamiento con distancia Mahalanobis.



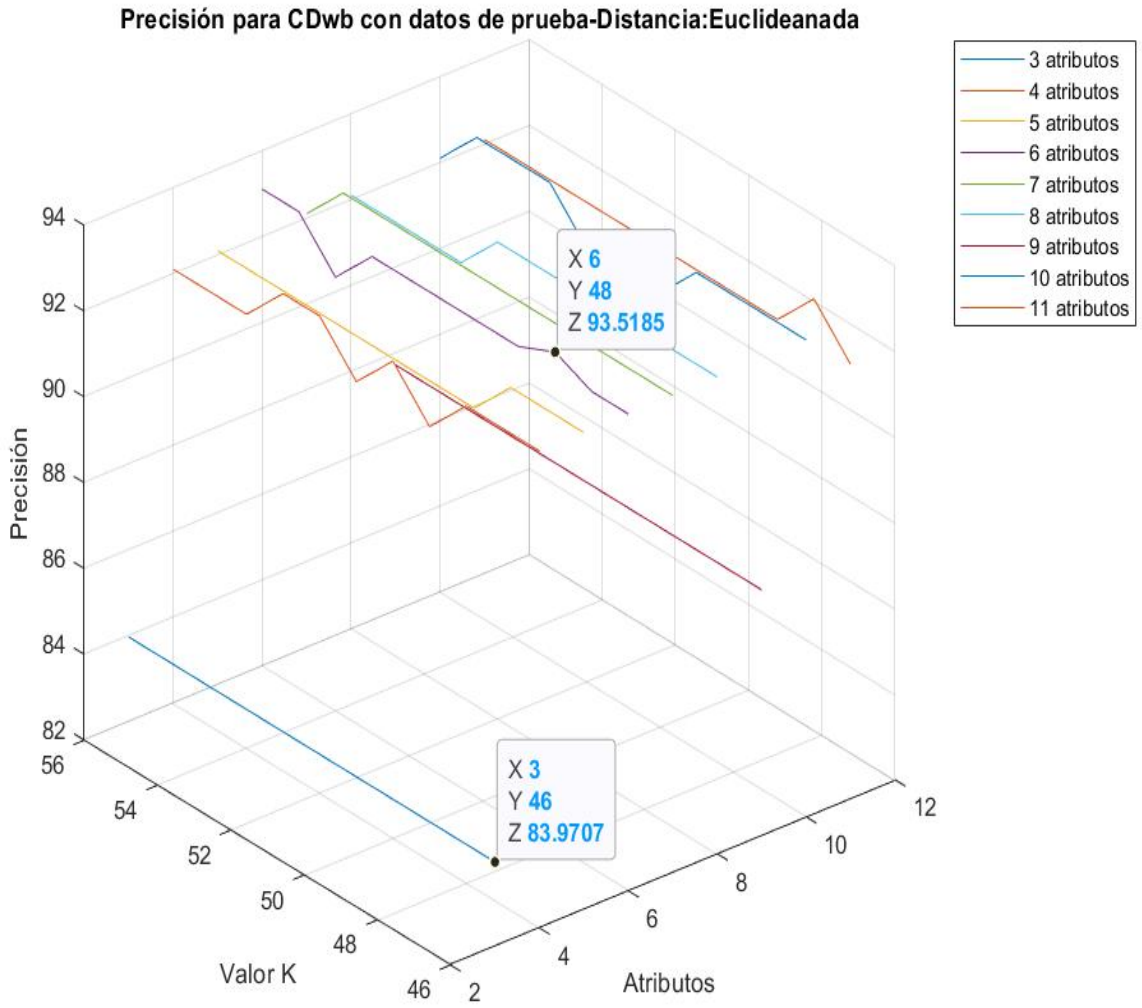
Nota: Obtenido de Autor.

9.2. Mejores resultados con datos de prueba

9.2.1. Resultados con distancia Euclidiana

Figura 55

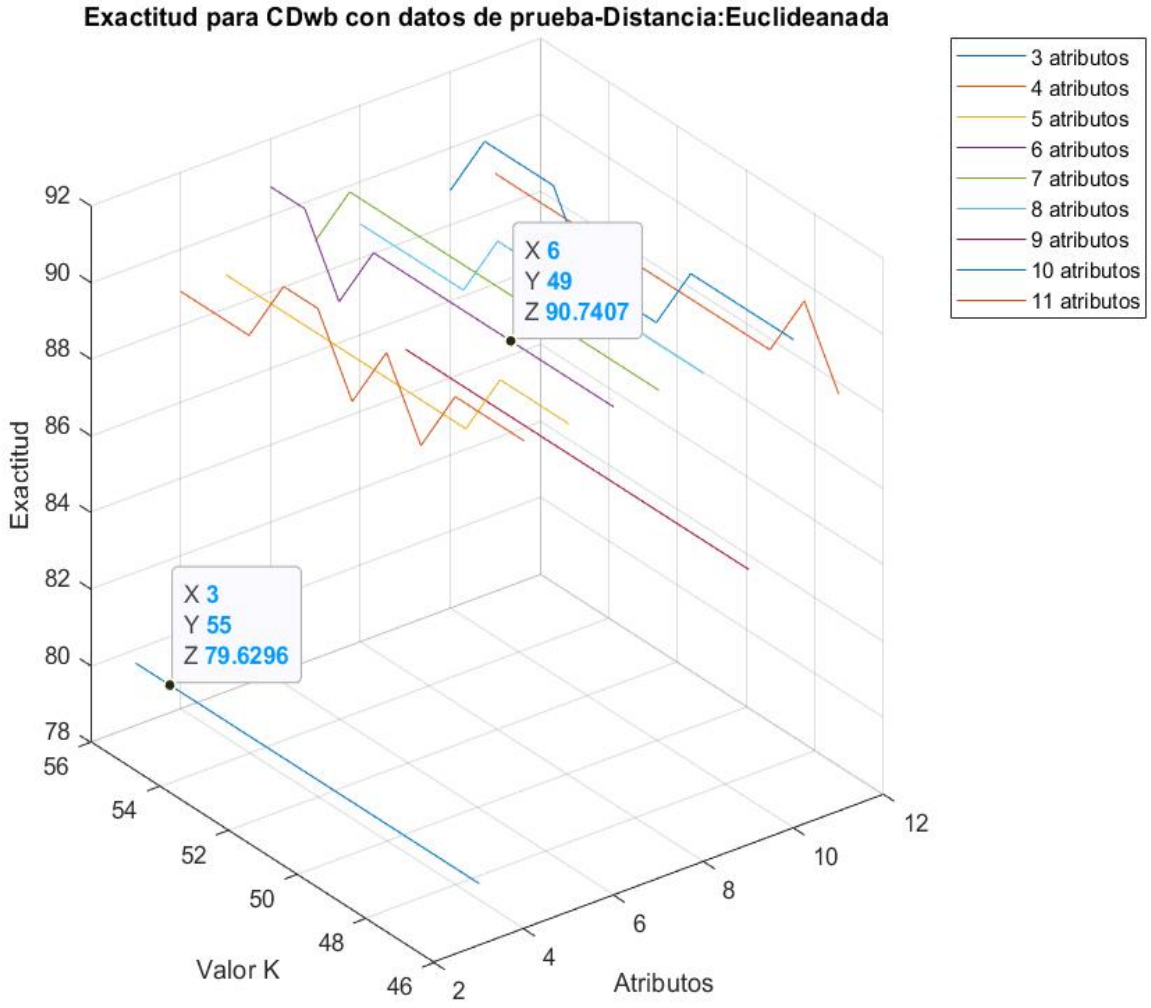
Precisión de CDbw con datos de prueba con distancia Euclidiana.



Nota: Obtenido de Autor.

Figura 56

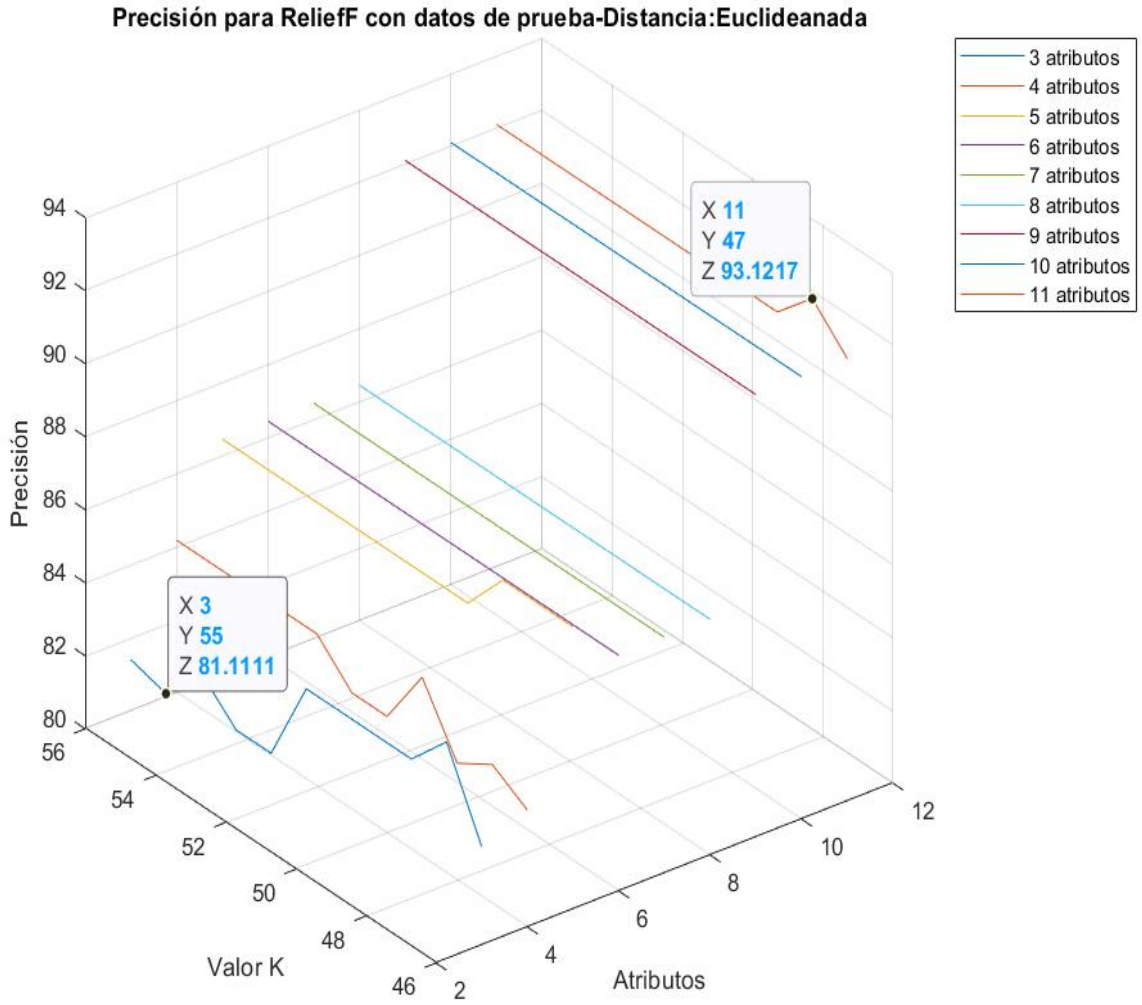
Exactitud de CDbw con datos de prueba con distancia Euclidiana.



Nota: Obtenido de Autor.

Figura 57

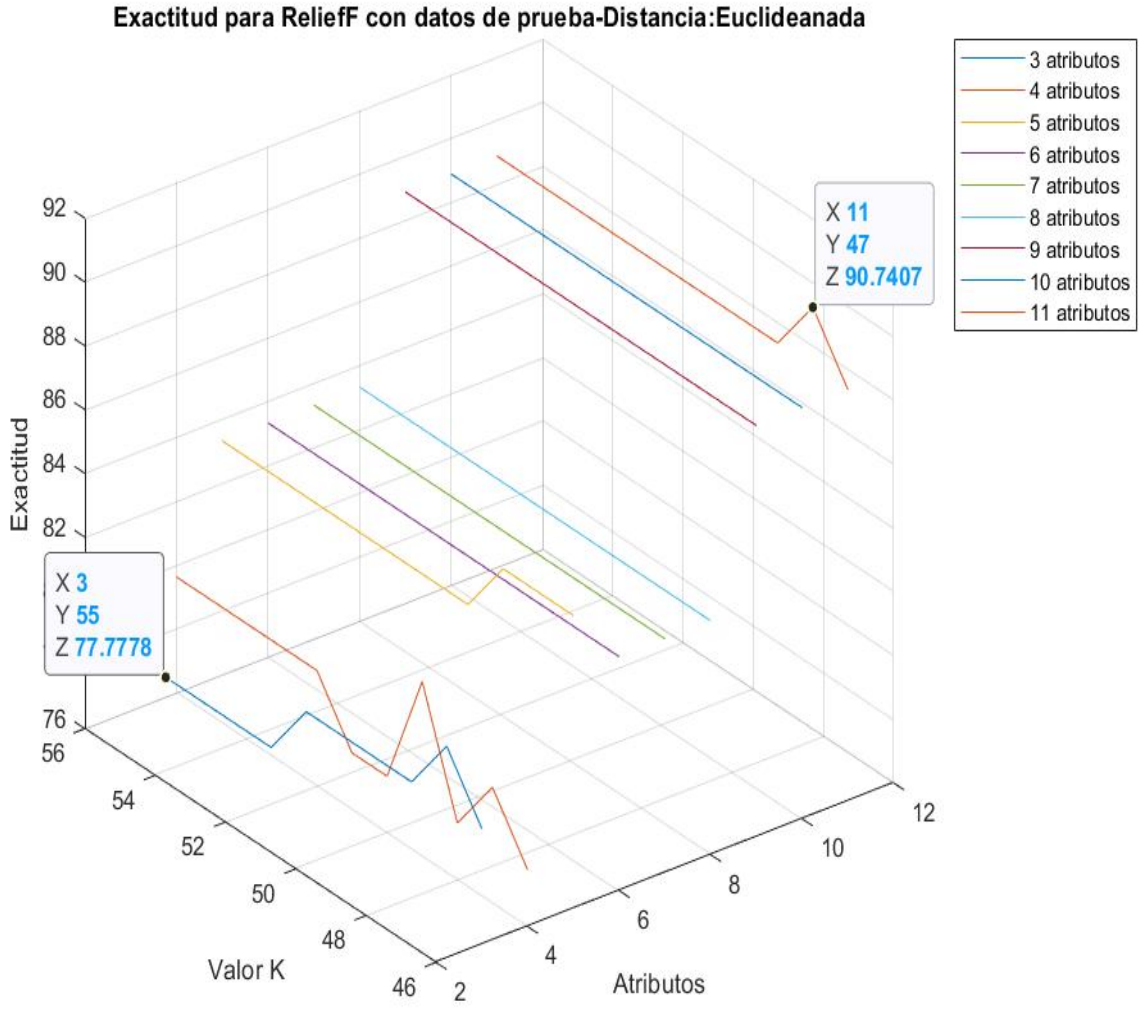
Precisión de ReliefF con datos de prueba con distancia Euclidiana.



Nota: Obtenido de Autor.

Figura 58

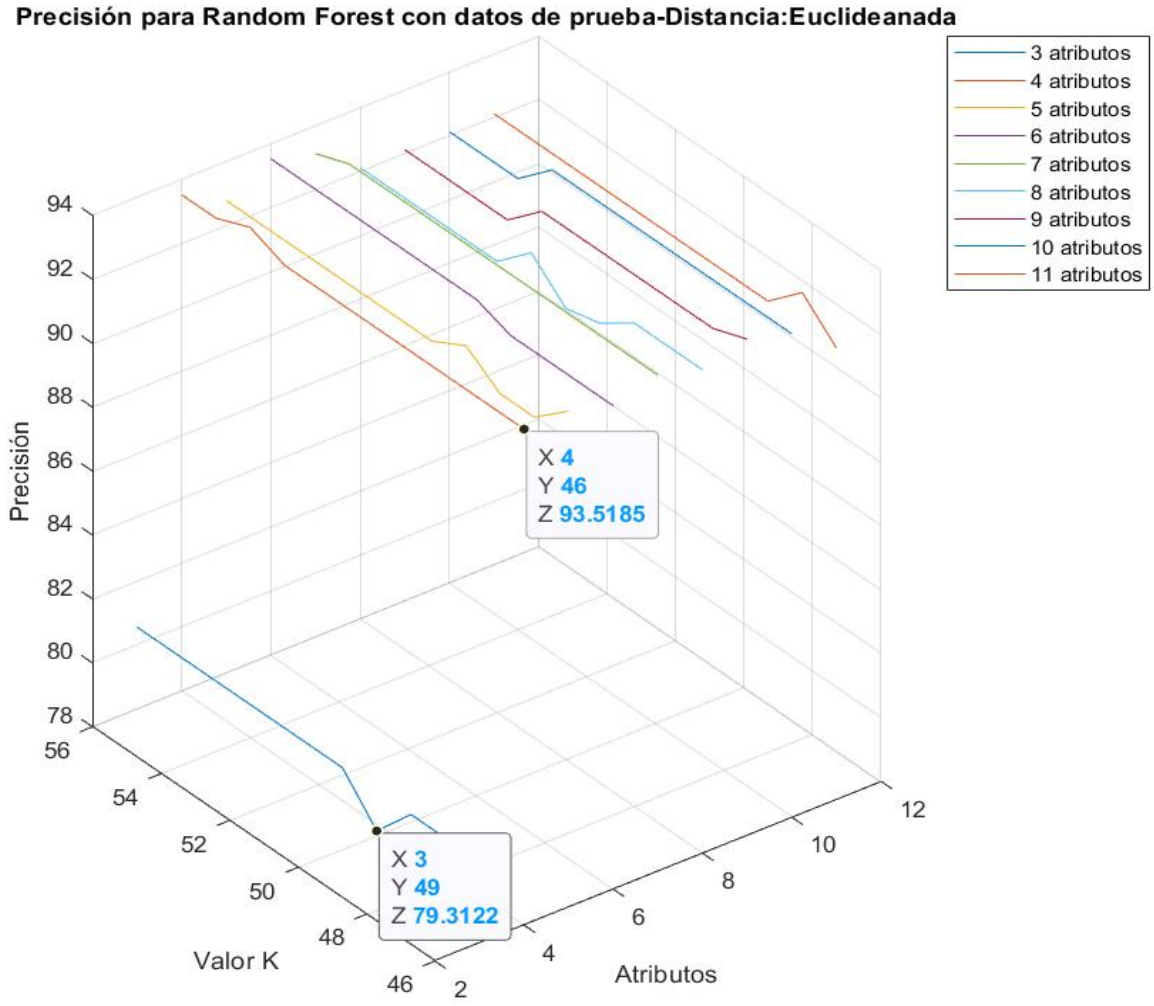
Exactitud de ReliefF con datos de prueba con distancia Euclidiana.



Nota: Obtenido de Autor.

Figura 59

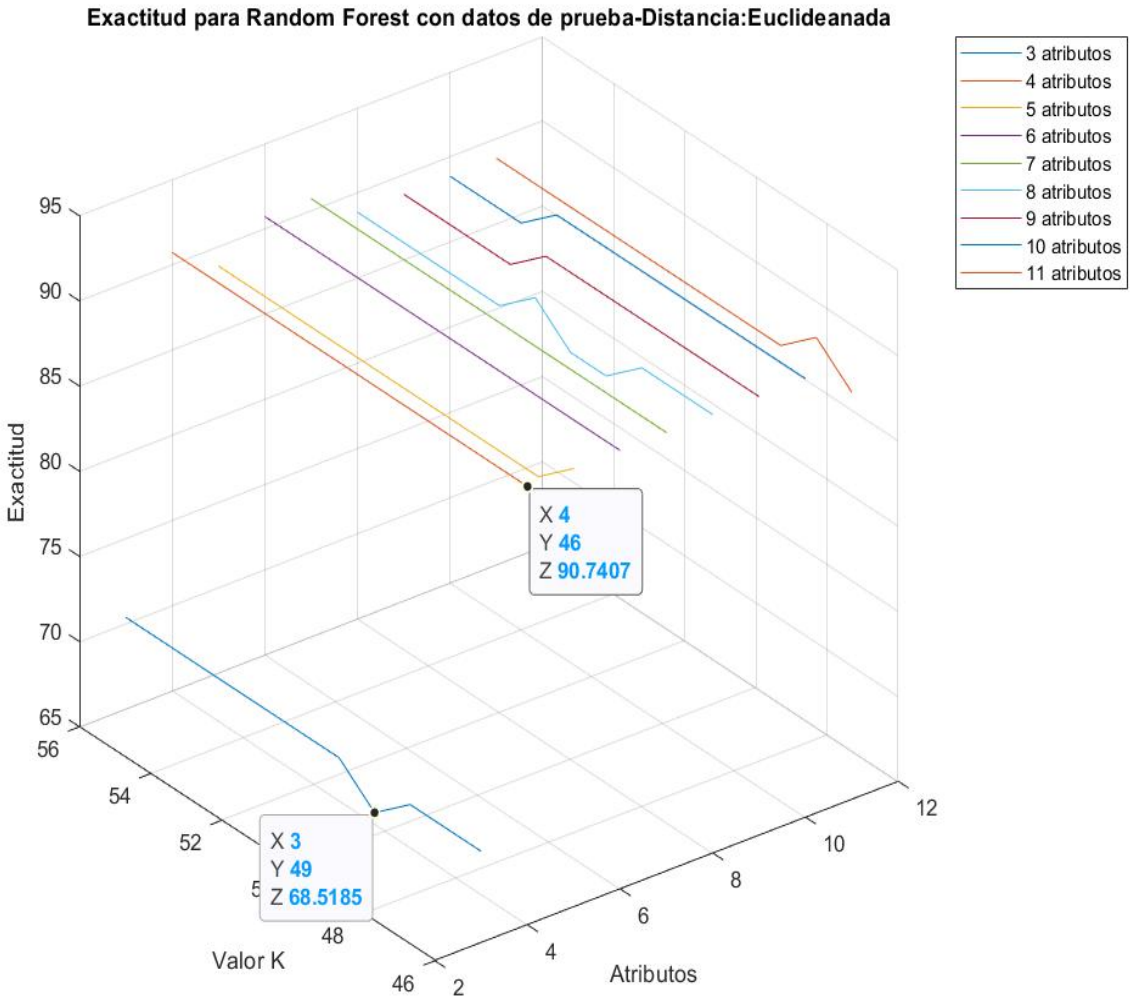
Precisión de Random Forest con datos de prueba con distancia Euclidiana.



Nota: Obtenido de Autor.

Figura 60

Exactitud de Random Forest con datos de prueba con distancia Euclidiana.

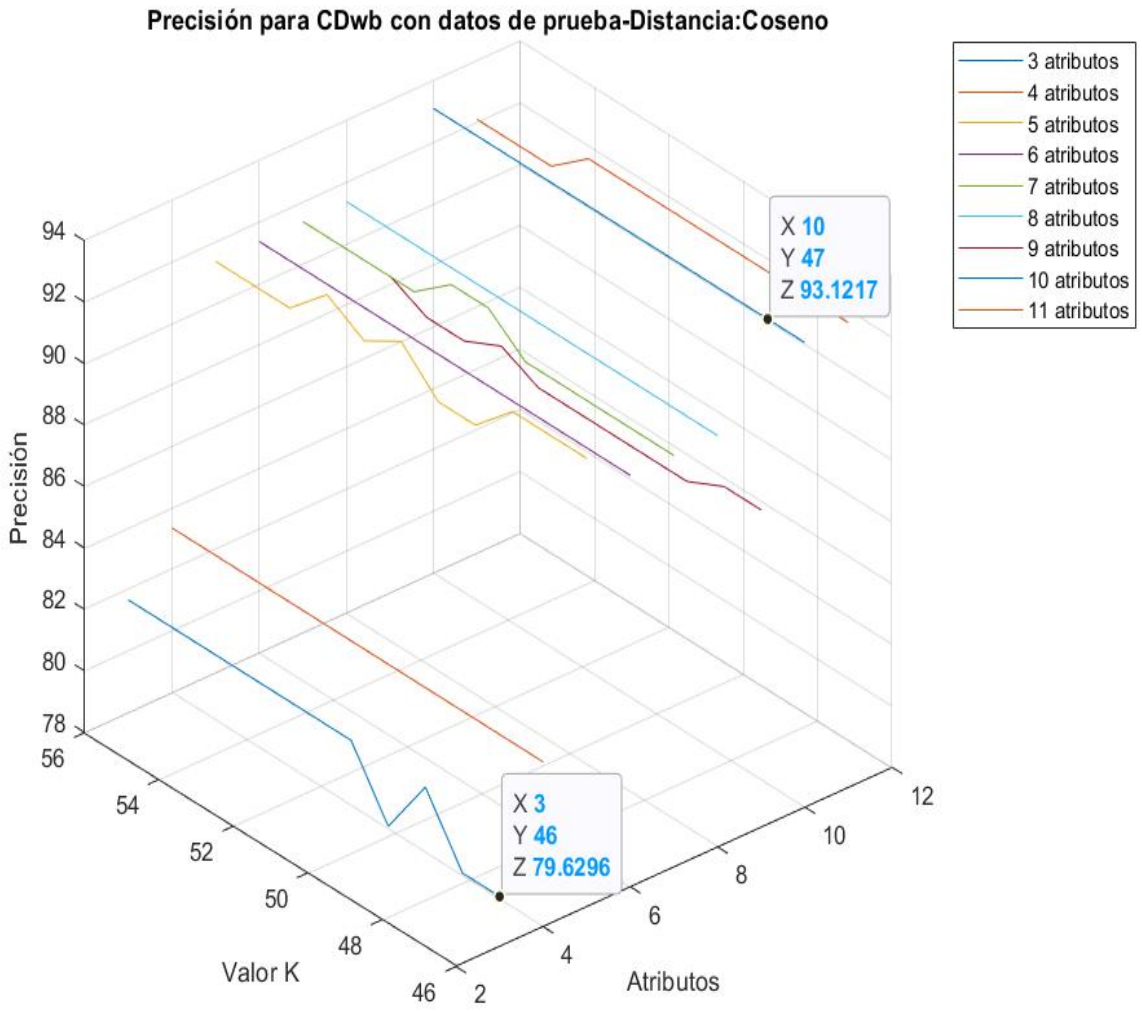


Nota: Obtenido de Autor.

9.2.2. Resultados con distancia Coseno

Figura 61

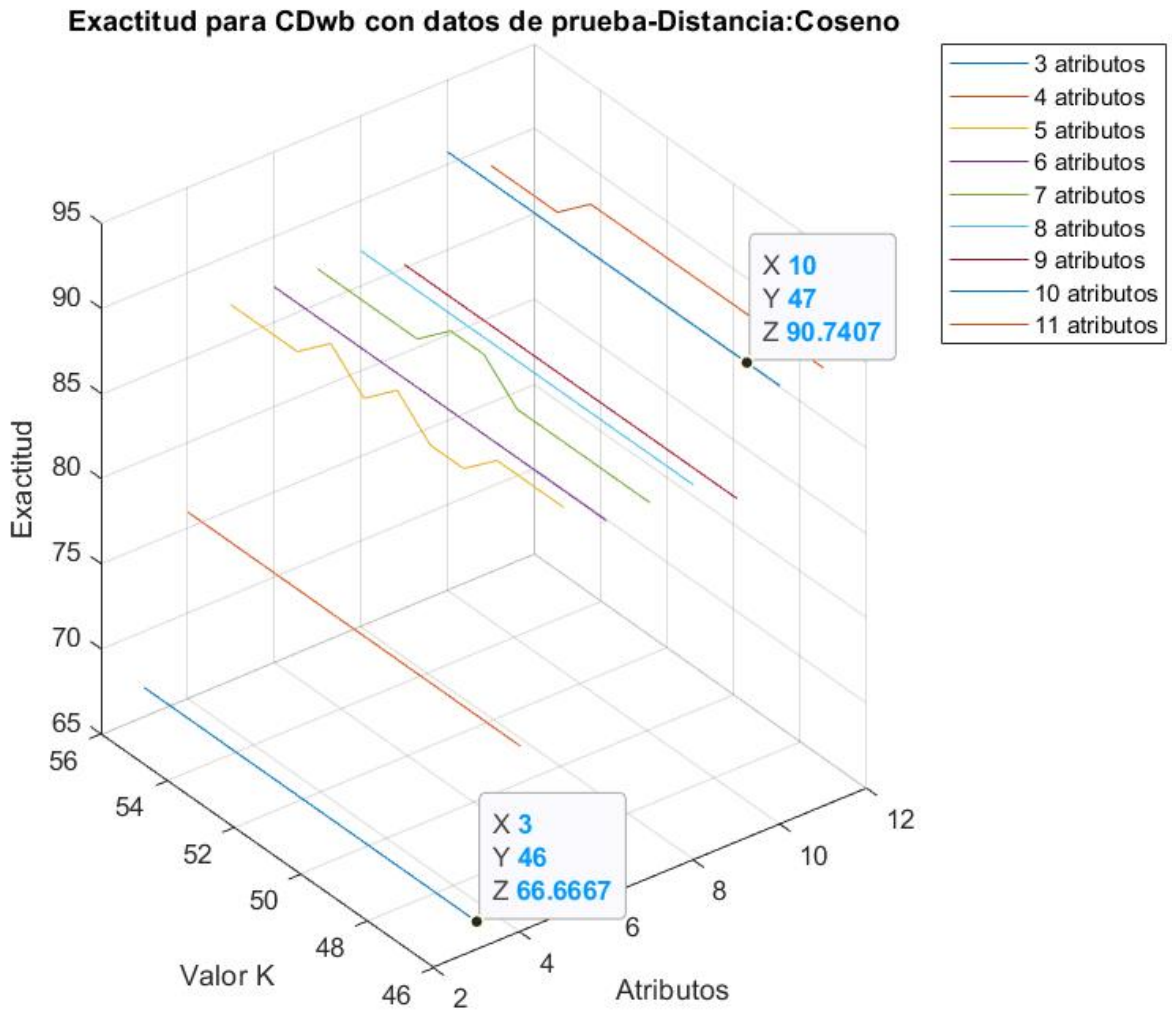
Precisión de CDbw con datos de prueba con distancia Coseno.



Nota: Obtenido de Autor.

Figura 62

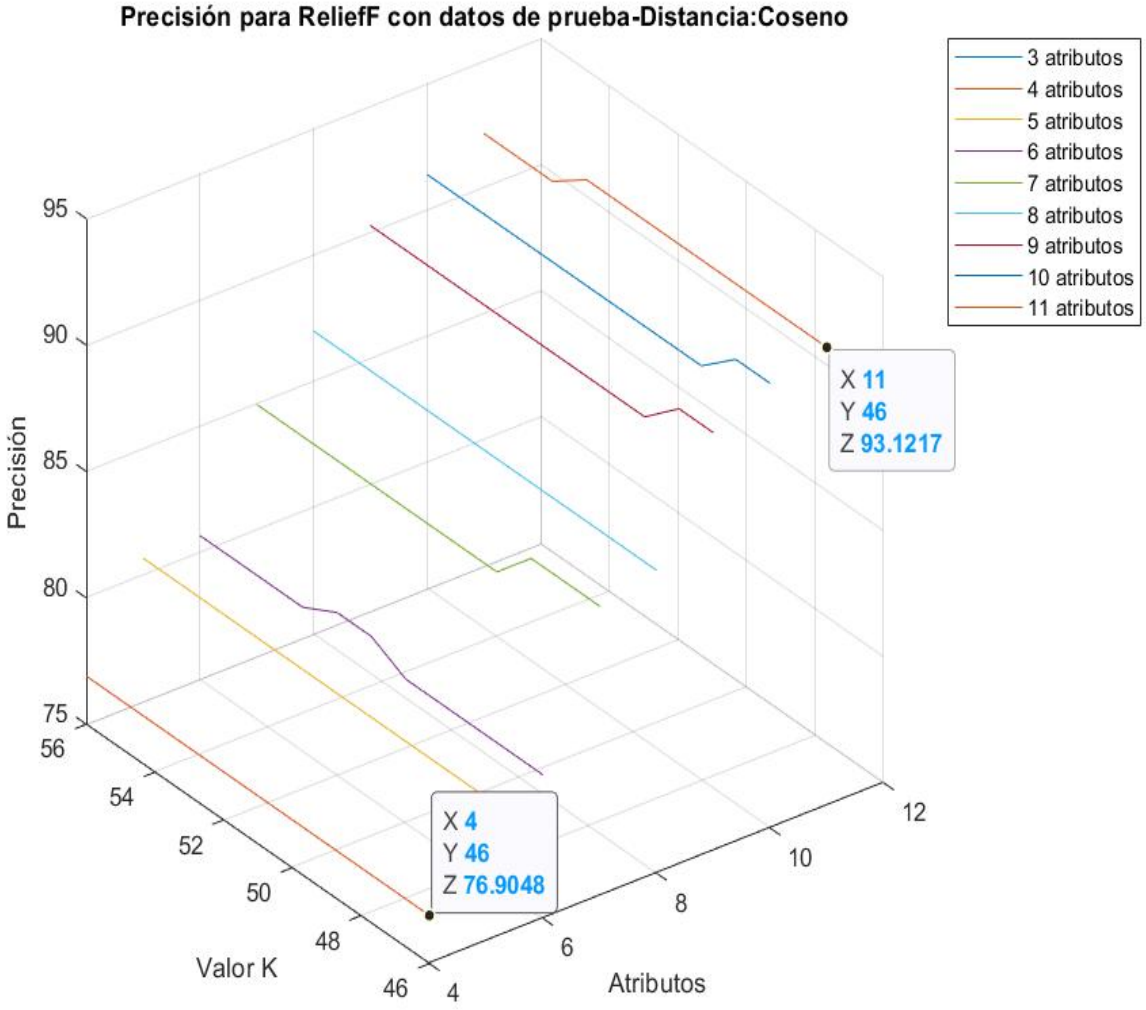
Exactitud de CDbw con datos de prueba con distancia Coseno.



Nota: Obtenido de Autor.

Figura 63

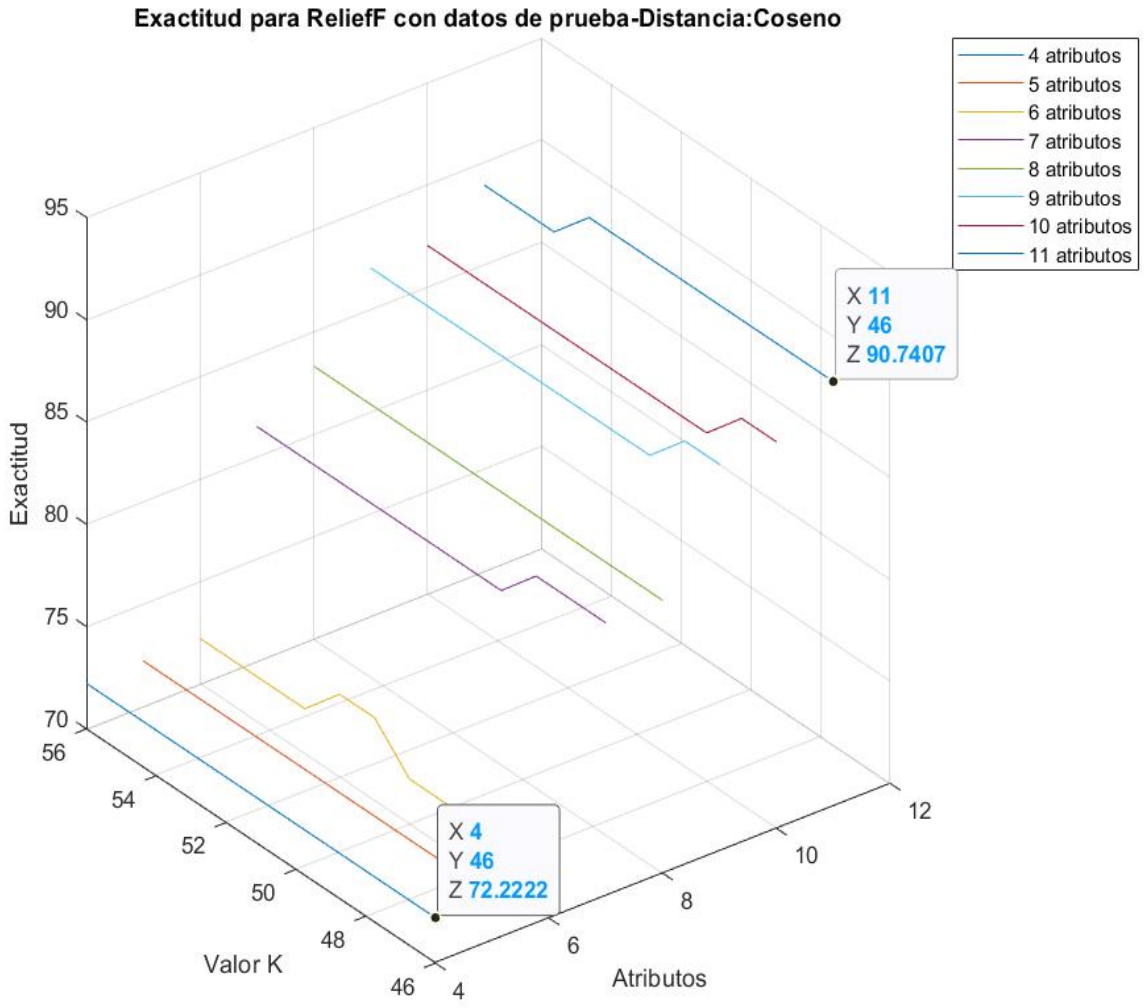
Precisión de ReliefF con datos de prueba con distancia Coseno.



Nota: Obtenido de Autor.

Figura 64

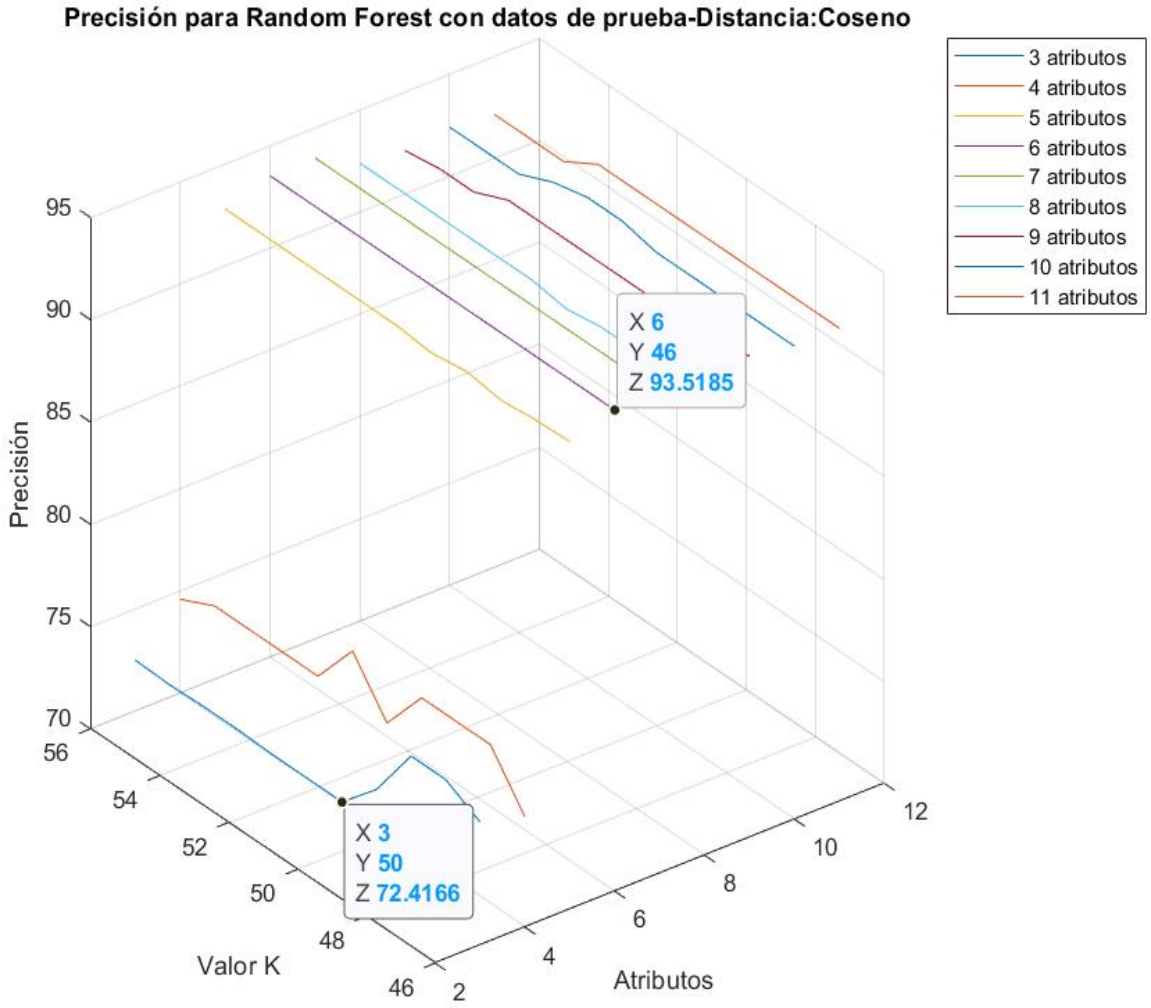
Exactitud de ReliefF con datos de prueba con distancia Coseno.



Nota: Obtenido de Autor.

Figura 65

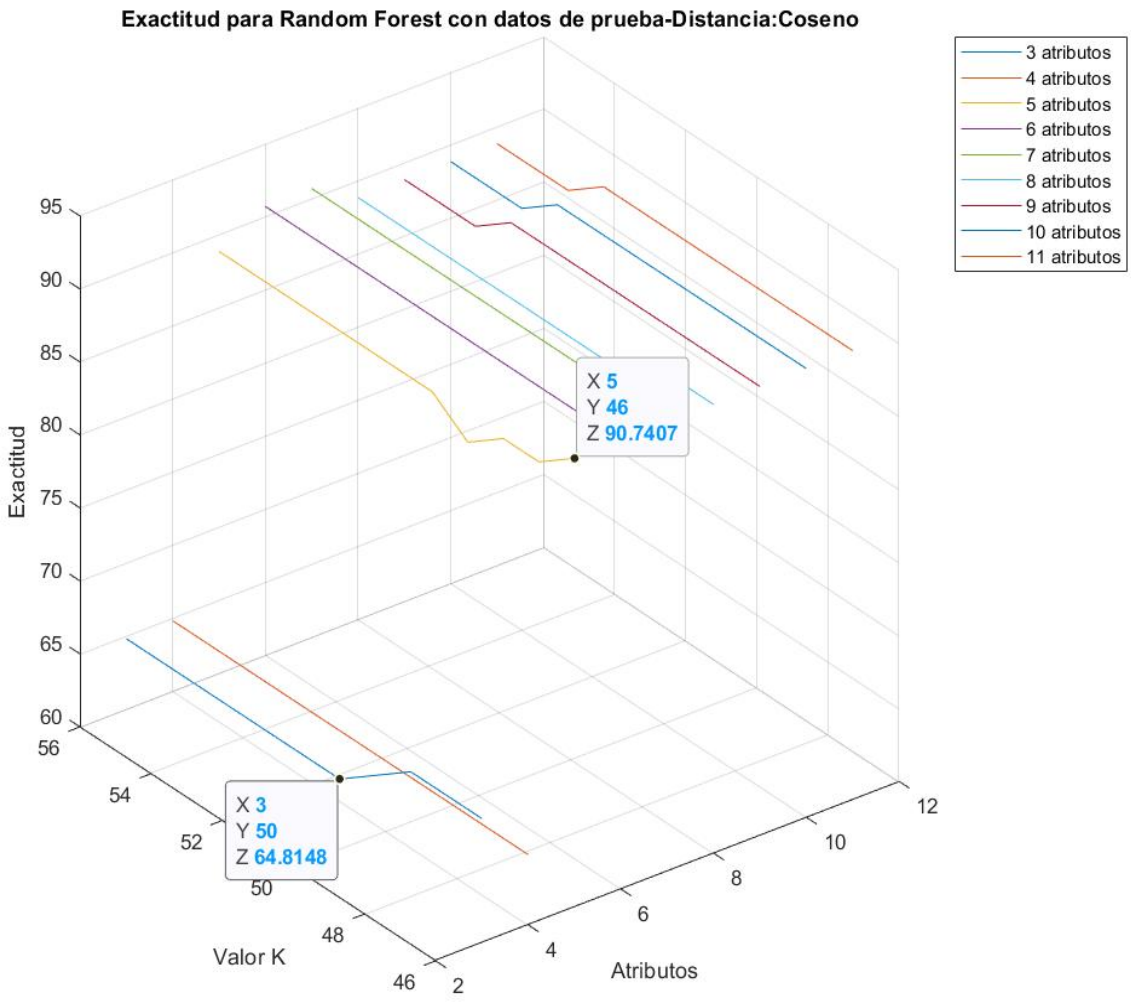
Precisión de Random Forest con datos de prueba con distancia Coseno.



Nota: Obtenido de Autor.

Figura 66

Exactitud de Random Forest con datos de prueba con distancia Coseno.

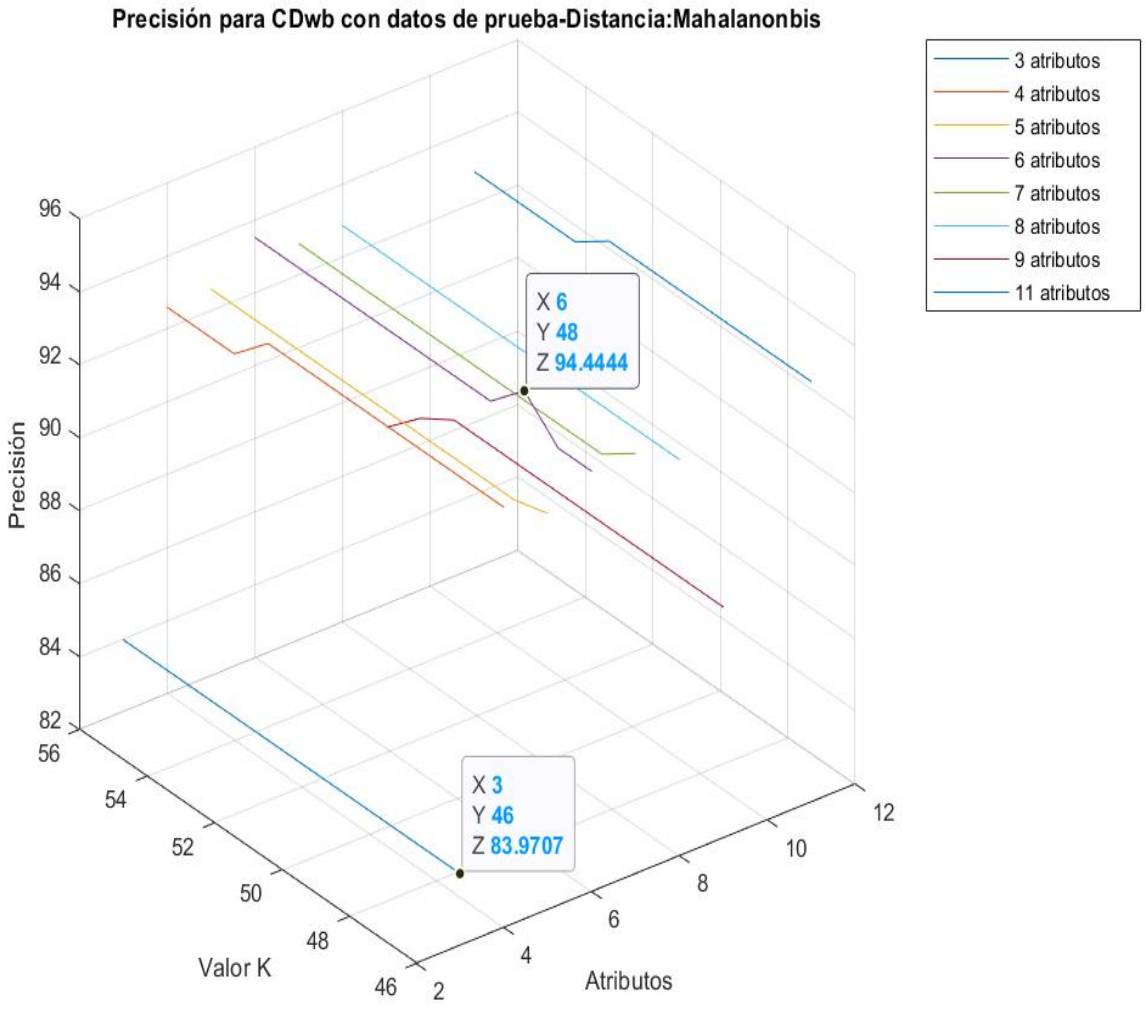


Nota: Obtenido de Autor.

9.2.3. Resultados con distancia Mahalanobis con datos de prueba

Figura 67

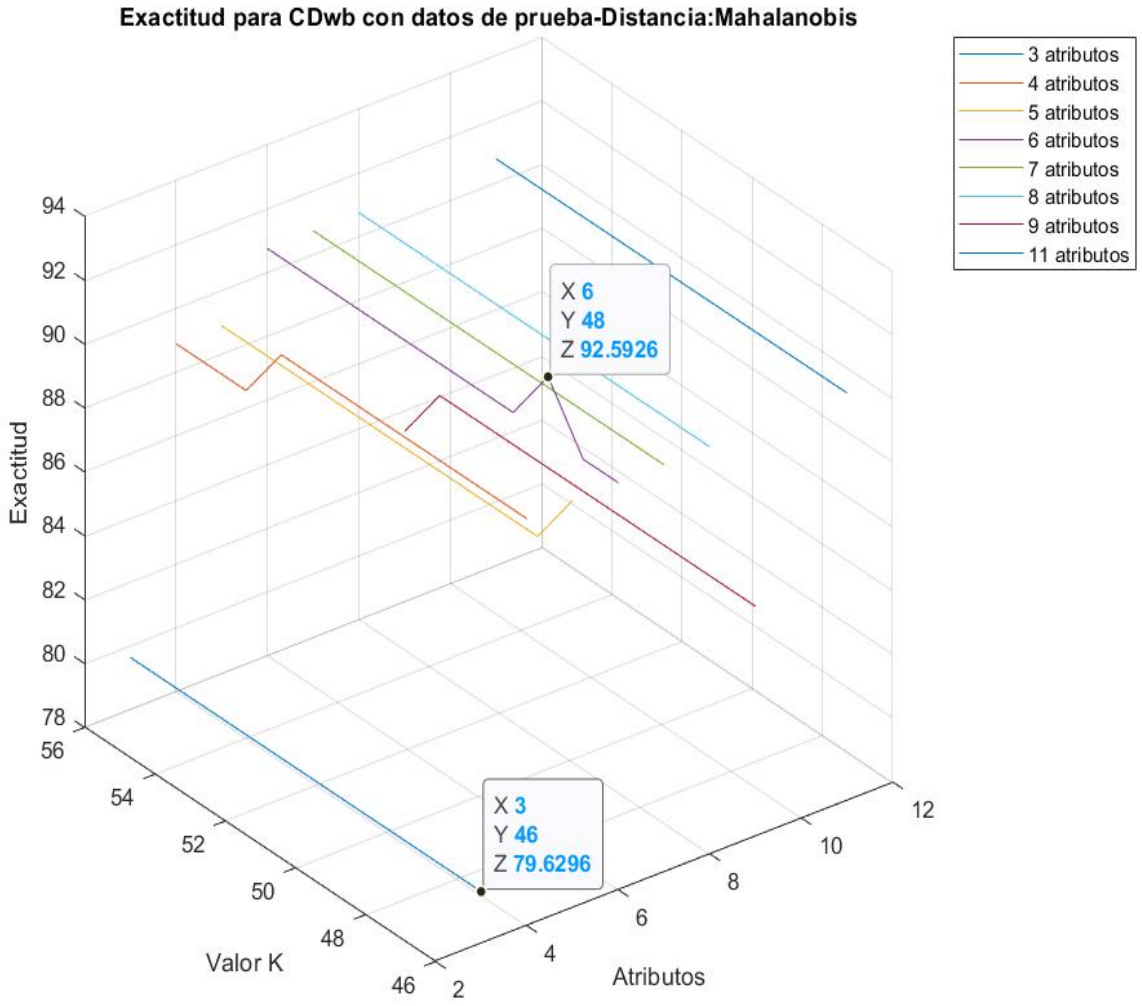
Precisión de CDbw con datos de de prueba con distancia Mahalanobis.



Nota: Obtenido de Autor.

Figura 68

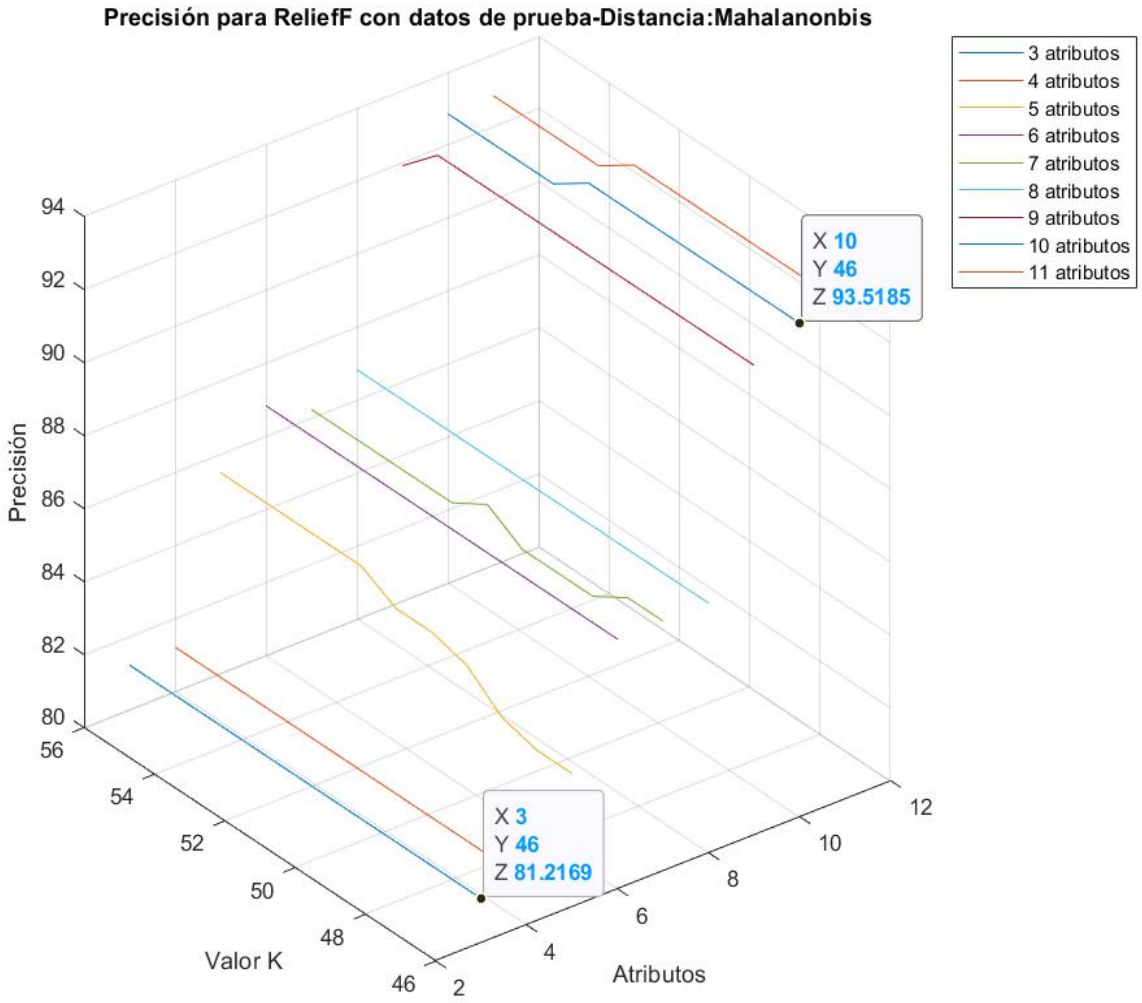
Exactitud de CDbw con datos de de prueba con distancia Mahalanobis.



Nota: Obtenido de Autor.

Figura 69

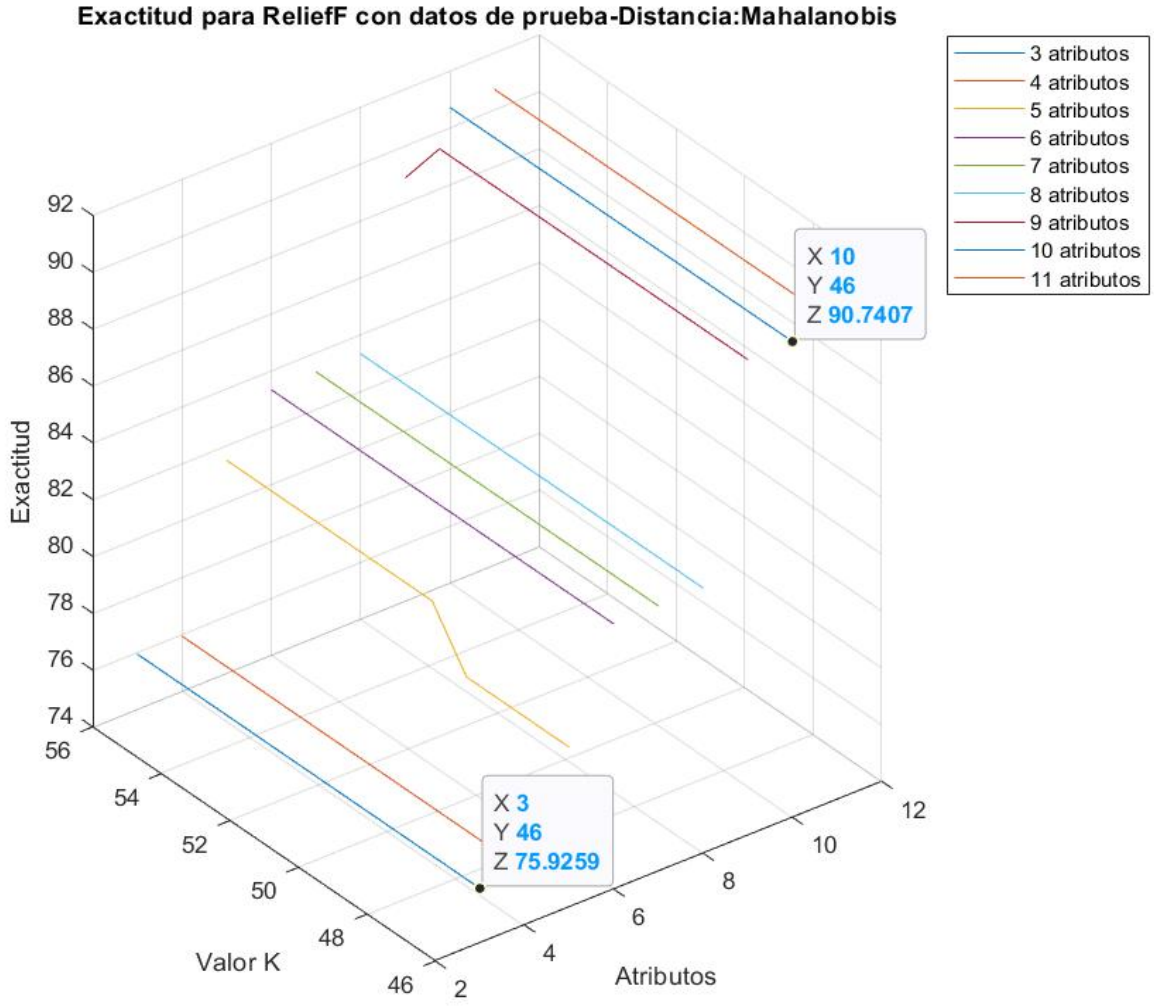
Precisión de ReliefF con datos de de prueba con distancia Mahalanobis.



Nota: Obtenido de Autor.

Figura 70

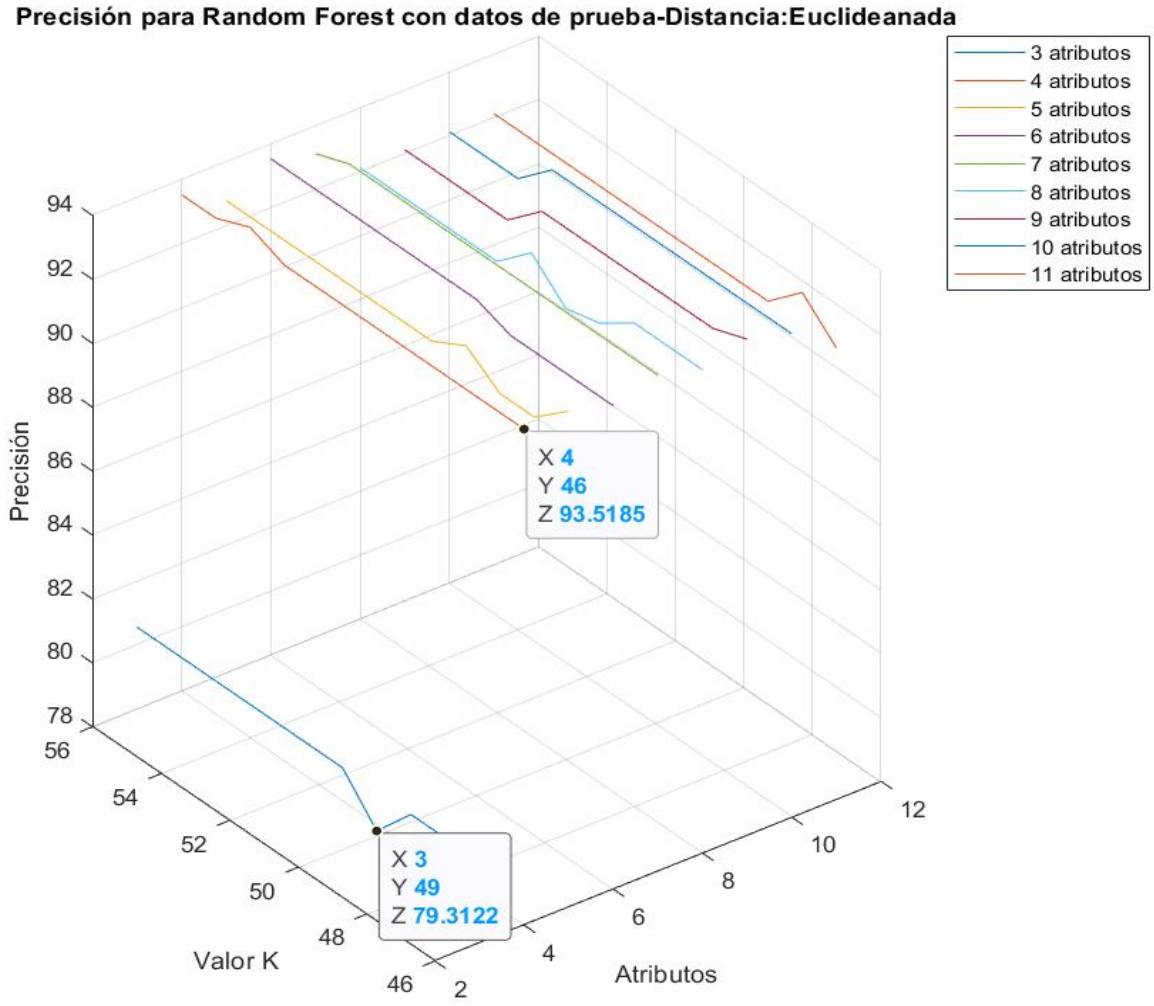
Exactitud de ReliefF con datos de de prueba con distancia Mahalanobis.



Nota: Obtenido de Autor.

Figura 71

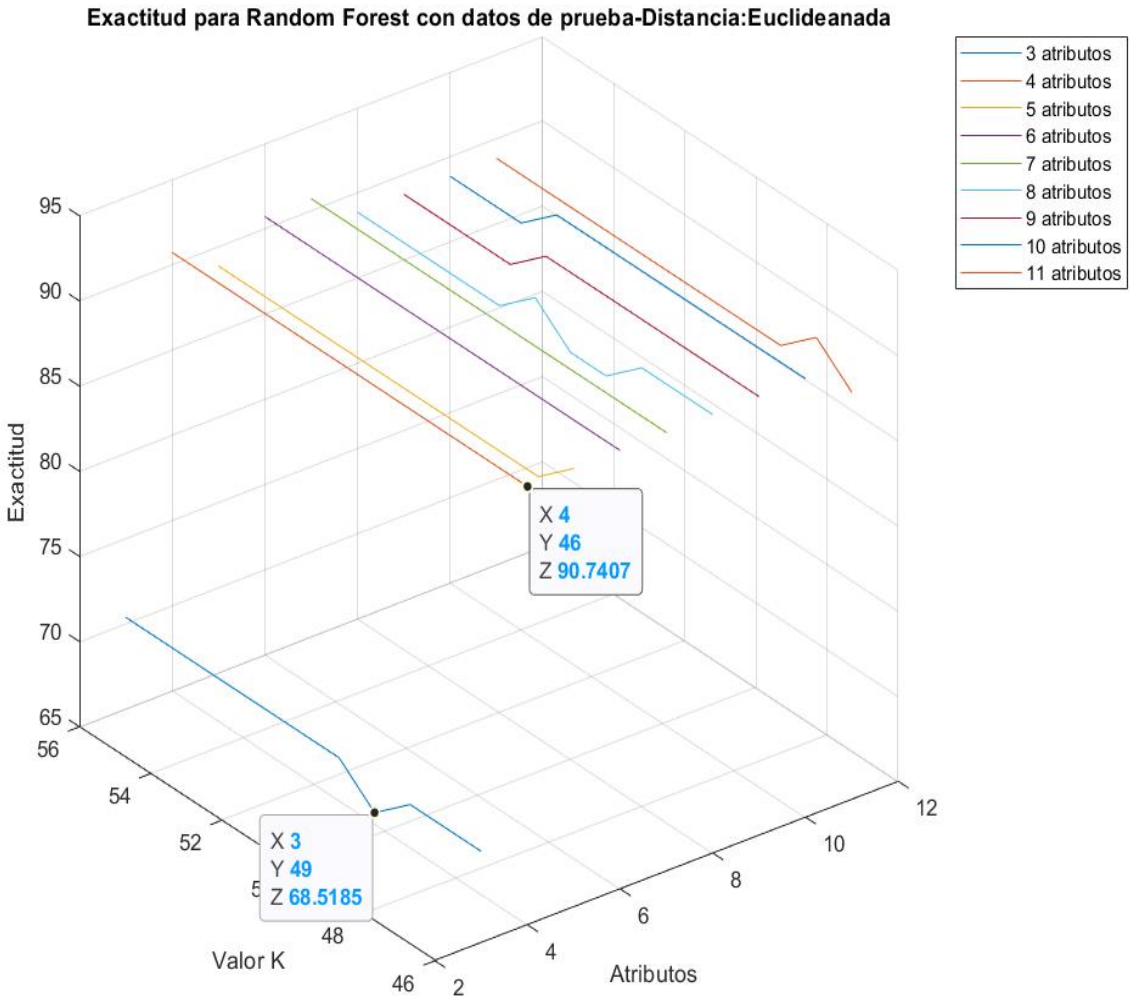
Precisión de Random Forest con datos de entrenamiento con distancia Mahalanobis.



Nota: Obtenido de Autor.

Figura 72

Exactitud de Random Forest con datos de entrenamiento con distancia Mahalanobis.

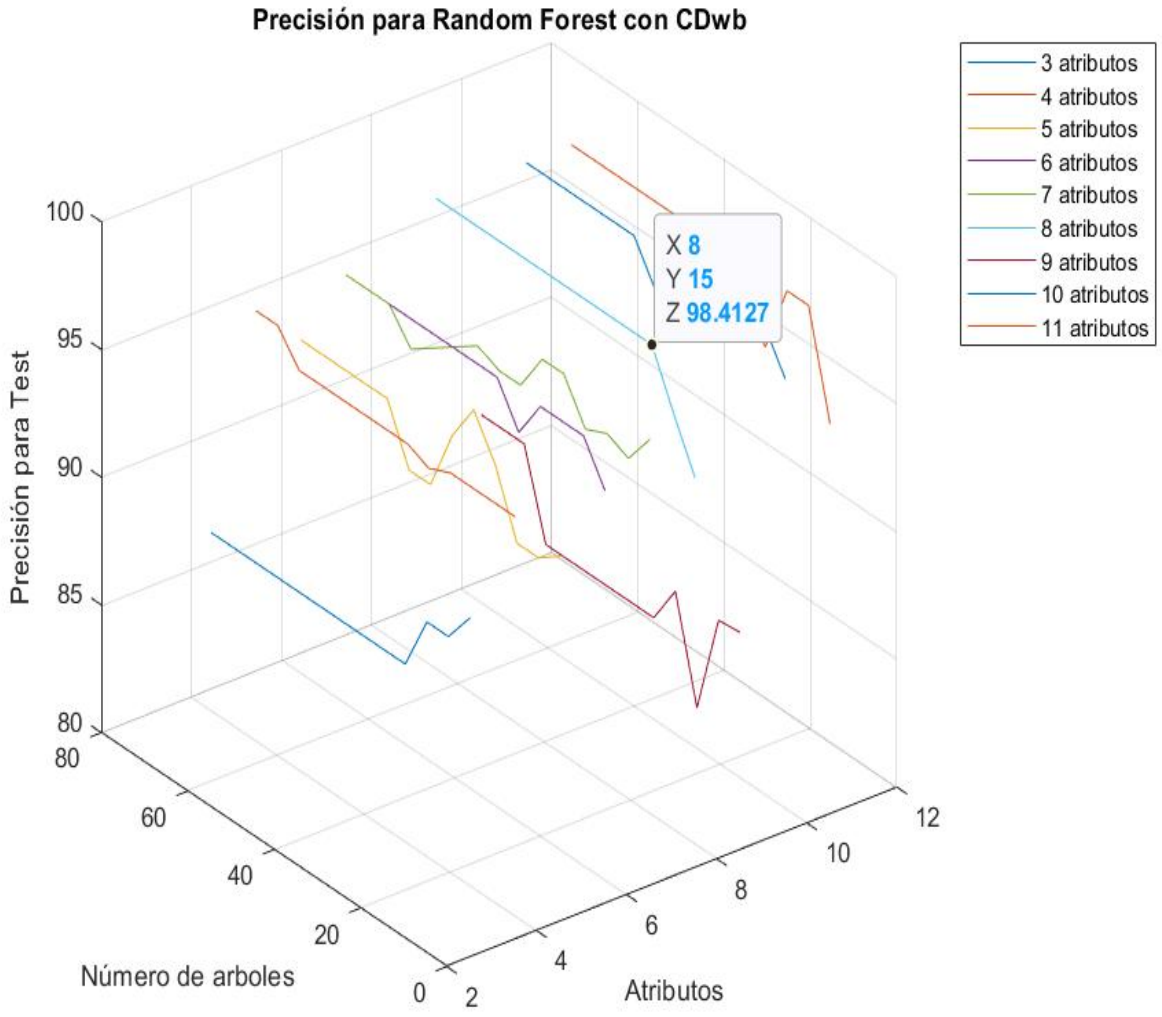


Nota: Obtenido de Autor.

9.3. Resultados con Random Forest

Figura 73

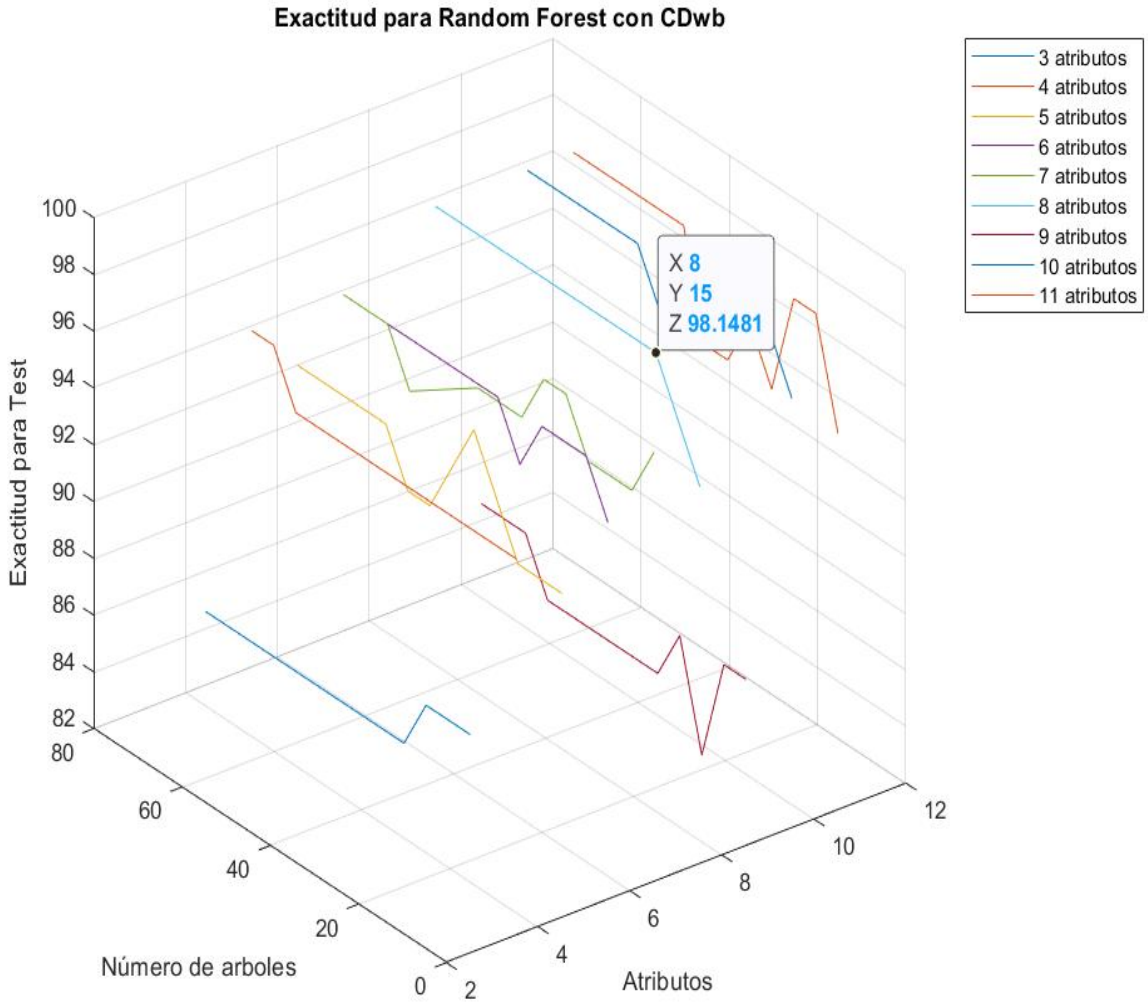
Precisión de Random Forest con datos de prueba usando CDbw.



Nota: Obtenido de Autor.

Figura 74

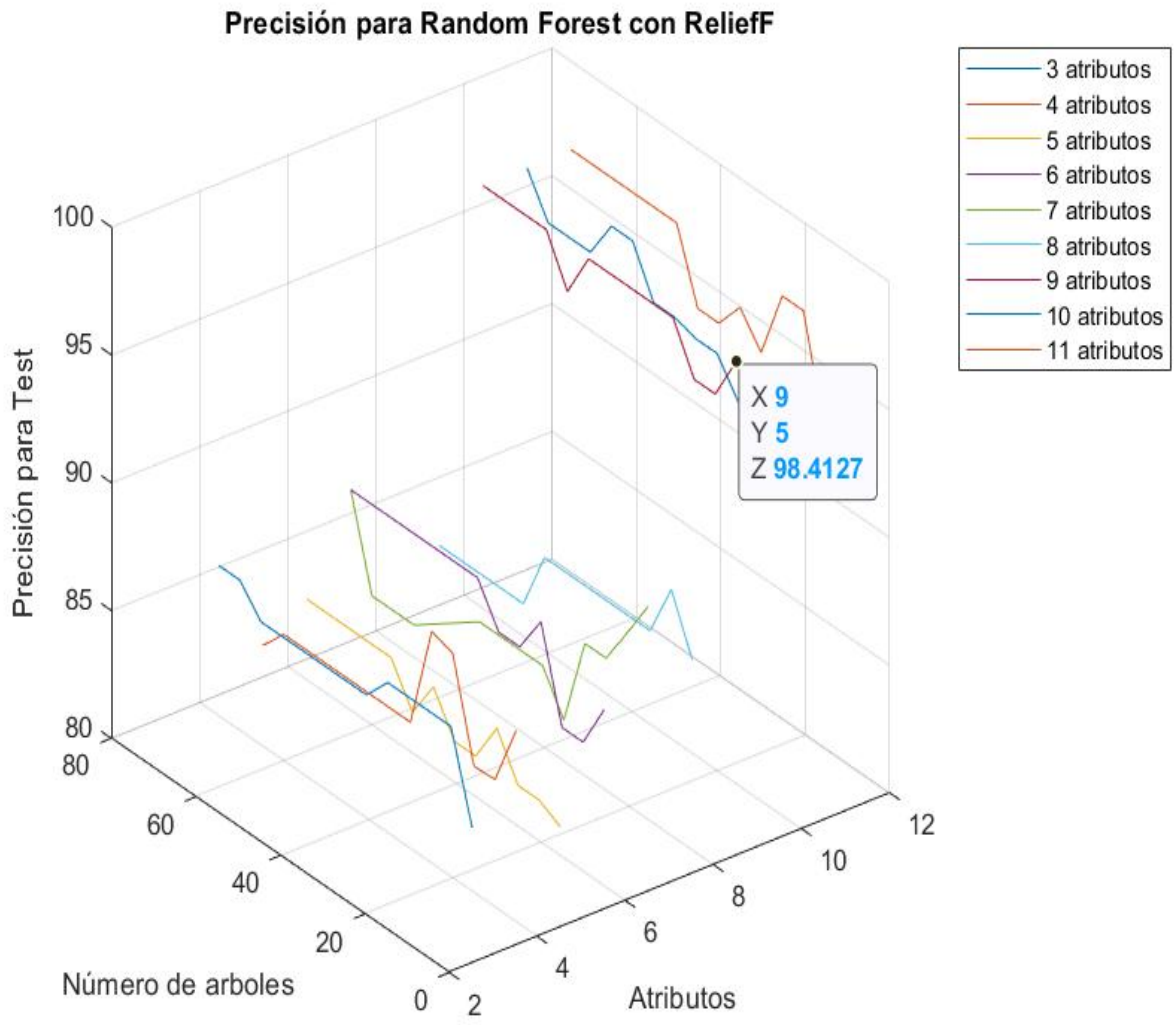
Exactitud de Random Forest con datos de prueba usando CDbw.



Nota: Obtenido de Autor.

Figura 75

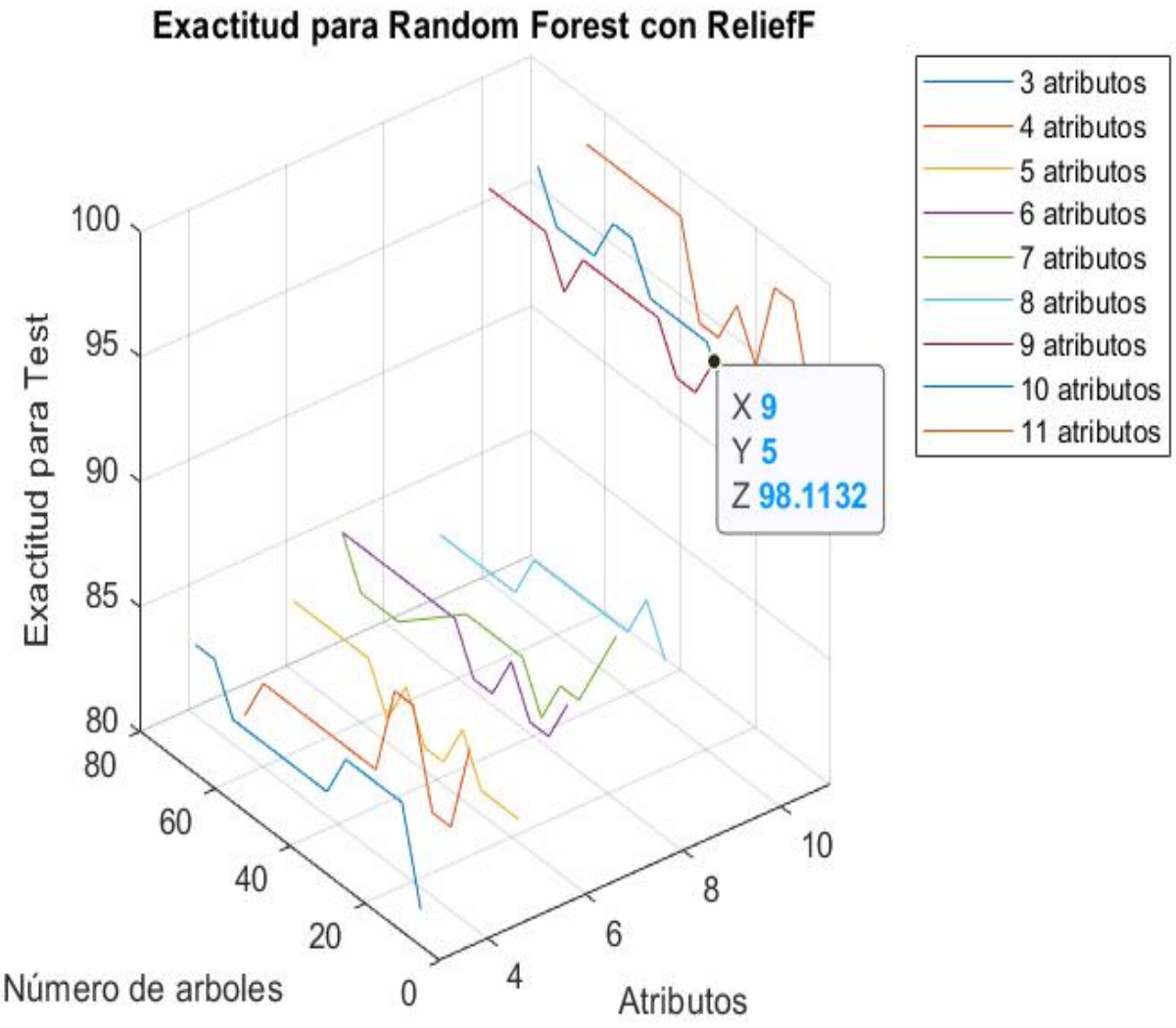
Precisión de Random Forest con datos de prueba usando ReliefF.



Nota: Obtenido de Autor.

Figura 76

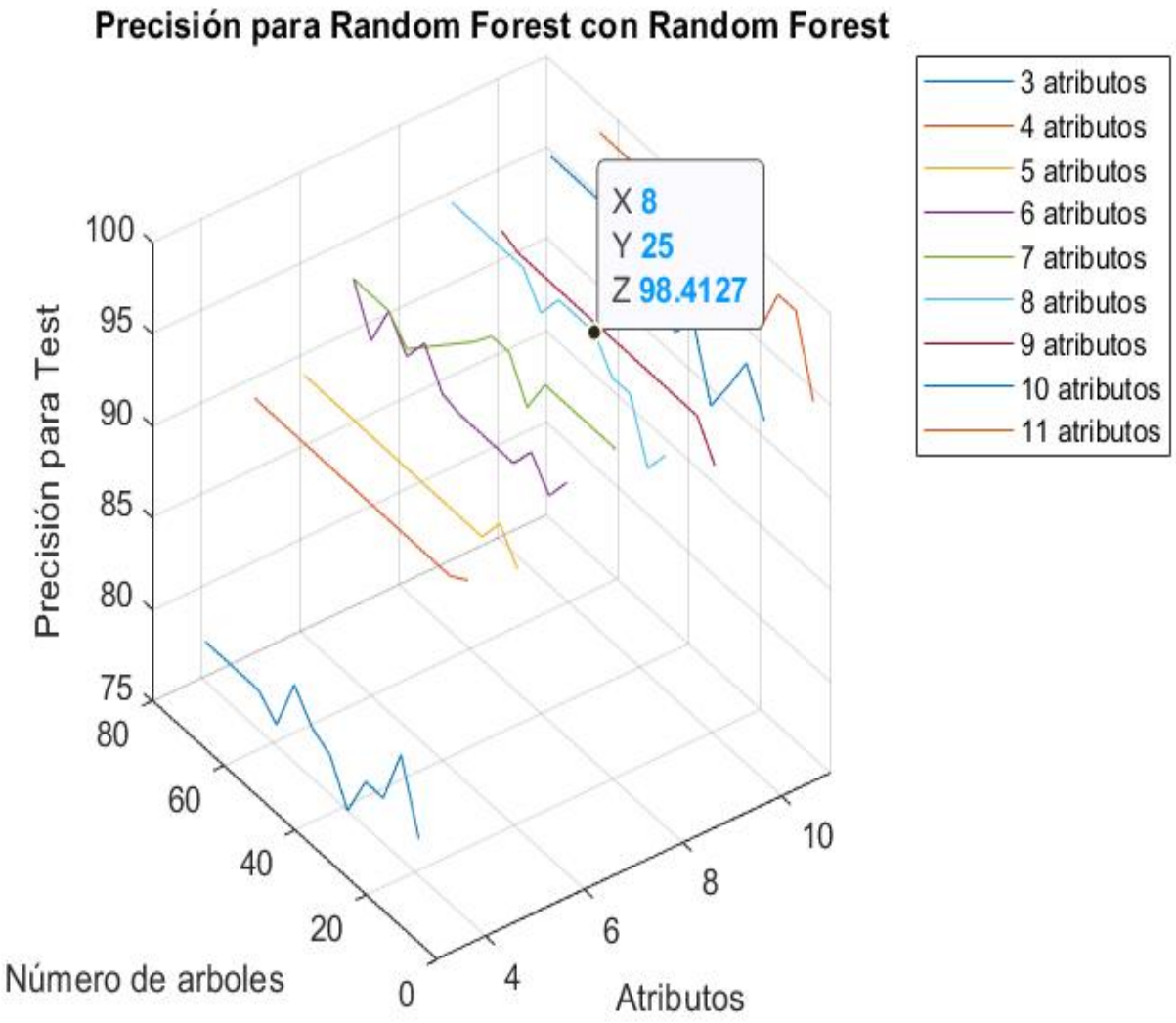
Exactitud de Random Forest con datos de prueba usando ReliefF.



Nota: Obtenido de Autor.

Figura 77

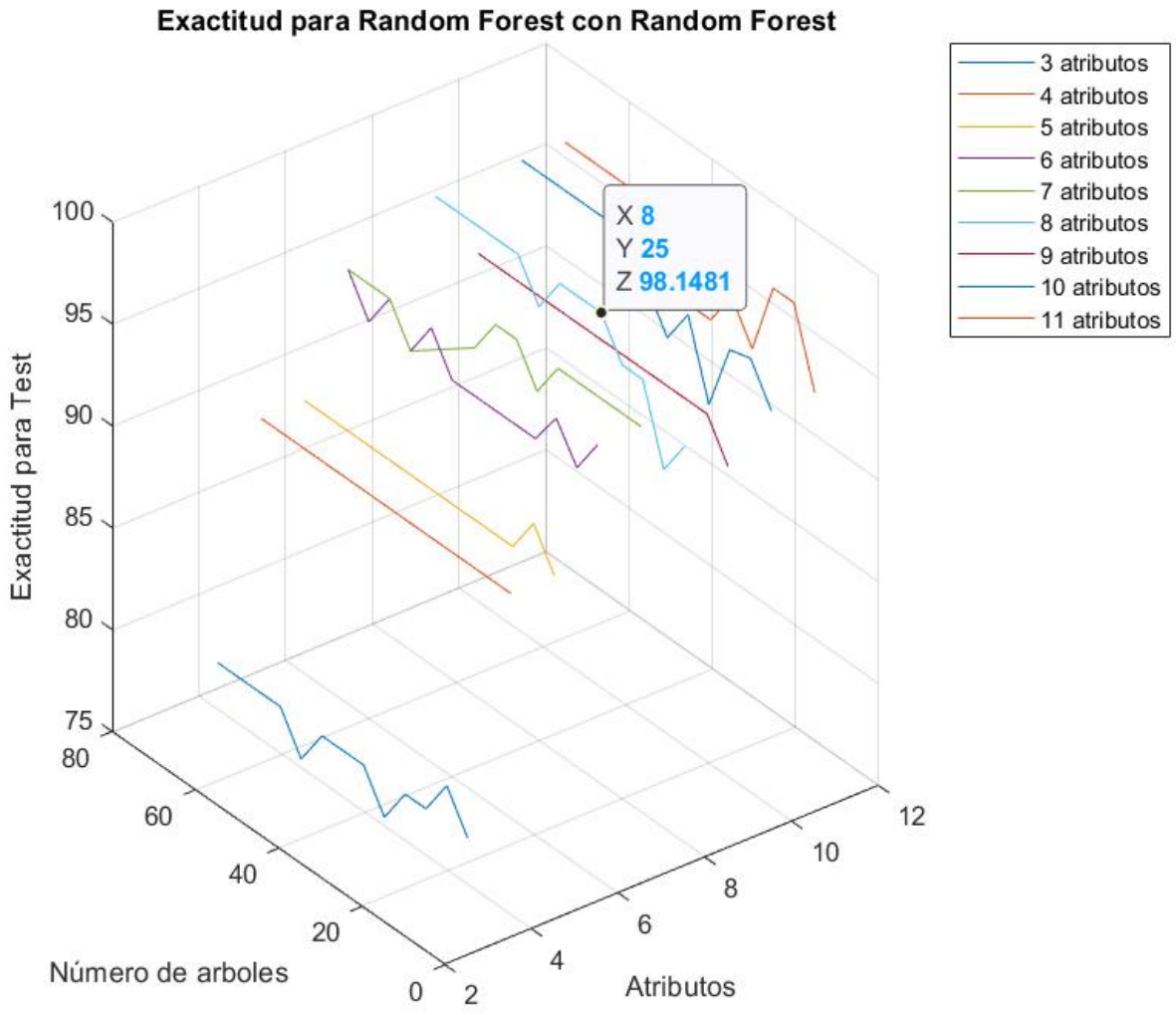
Precisión de Random Forest con datos de prueba usando Random Forest.



Nota: Obtenido de Autor.

Figura 78

Exactitud de Random Forest con datos de prueba usando Random Forest.



Nota: Obtenido de Autor.