

61

Fecha de presentación: enero, 2023

Fecha de aceptación: marzo, 2023

Fecha de publicación: mayo, 2023

EL ENFOQUE DE APRENDIZAJE

DE MÁQUINA PARA LA GESTIÓN DEL MANTENIMIENTO INDUSTRIAL

THE MACHINE LEARNING APPROACH TO INDUSTRIAL MAINTENANCE MANAGEMENT

Karina Lemache-Caiza¹

E-mail: karina2032010@hotmail.com

ORCID: <https://orcid.org/0000-0002-2797-9218>

Félix García-Mora²

E-mail: felix.garcia@esPOCH.edu.ec

ORCID: <https://orcid.org/0000-0001-5814-3694>

Vanessa Valverde-González²

E-mail: v_valverde@esPOCH.edu.ec

ORCID: <https://orcid.org/0000-0002-6762-768X>

Efraín Velastegui López³

E-mail: evelasteguil@utb.edu.ec

ORCID: <https://orcid.org/0000-0002-7353-5853>

¹Investigador independiente. Ecuador.

²Escuela Superior Politécnica de Chimborazo (ESPOCH). Ecuador.

³Universidad Técnica de Babahoyo. Ecuador.

Cita sugerida (APA, séptima edición)

Lemache-Caiza, K., García-Mora, F., Valverde-González, V. & Velastegui López, E. (2023). El enfoque de aprendizaje de máquina para la gestión del mantenimiento industrial. *Universidad y Sociedad*, 15(3), 628-637.

RESUMEN

La manufactura inteligente y la innovación de la industria 4.0 a nivel mundial forman parte de la transformación tecnológica para crear sistemas de gestión y formas de hacer negocios, que permitan optimizar los procesos de fabricación, alcanzar una mayor flexibilidad y eficiencia, así como responder de forma oportuna a las necesidades de su mercado. El aprendizaje de máquina constituye una tecnología que es capaz de predecir con fiabilidad determinados resultados a partir de un modelo preparado entrenándolo con datos de entrada anteriores y su comportamiento de salida. La investigación realizada se propuso como objetivo la comparación de modelos de aprendizaje de máquina para la detección de fallas en turborreactores de doble flujo extraído del Repositorio del Centro de Excelencia de Pronósticos de la NASA. Los resultados obtenidos se comparan con datos reales para verificar la exactitud dando como resultado el algoritmo Random Forest como el mejor modelo ejecutado con parámetros normales y optimizados con un f1-score de 99.949% y 99.99% respectivamente. Finalmente se conoce que en la base de datos no es posible realizar una extracción fiable y válida de las principales características mediante el aprendizaje automático, debido a su particularidad en las condiciones operativas. También es importante mencionar que el modelo SVM no fue ejecutado con hiperparámetros. Es recomendable utilizar métodos de comparación de aprendizaje profundo por su precisión al momento de clasificar los datos reduciendo drásticamente la carga computacional al momento de ejecutar el modelo.

Palabras clave: aprendizaje de máquina, turborreactor de doble flujo, mantenimiento industrial

ABSTRACT

Smart manufacturing and Industry 4.0 innovation worldwide are part of the technological transformation to create management systems and ways of doing business that optimize manufacturing processes, achieve greater flexibility and efficiency, and respond in a timely manner to the needs of their market. Machine learning is a technology that is able to reliably predict certain outcomes from a prepared model by training it with previous input data and its output behavior. The research carried out was aimed at comparing machine learning models for the detection of failures in twin-flow turbojets extracted from

the NASA Prediction Centre of Excellence Repository. The results obtained are compared with real data to verify the accuracy resulting in the Random Forest algorithm as the best model run with normal and optimized parameters with an f1-score of 99.949% and 99.99% respectively. Finally, it is known that in the database it is not possible to perform a reliable and valid extraction of the main features by machine learning, due to its particularity in the operating conditions. It is also important to mention that the SVM model was not run with hyperparameters. It is advisable to use deep learning matching methods because of their accuracy in classifying the data and drastically reducing the computational load when running the model.

KEYWORDS: machine learning, twin-flow turbojet, industrial maintenance, industrial maintenance

INTRODUCCIÓN

Un turborreactor de doble flujo es un propulsor que funciona como un motor a reacción, donde la primera etapa de compresión es reemplazada por un gran ventilador que proporciona la mayor parte del empuje y es más comúnmente utilizado en motores de vuelo comercial debido a su eficiencia y bajo consumo de combustible frente a otras opciones (Gordón, 2017).

La necesidad de monitorear constantemente el estado de la propiedad física y la presencia de una gran cantidad de datos industriales, refleja que aplicar Machine Learning es la mejor opción para lograr un diagnóstico automático y confiable acerca del estado de las máquinas, además la disponibilidad de estrategias de mantenimiento causadas por máquinas le permitirá obtener grandes ventajas en el rendimiento, seguridad, optimización y toma de decisiones; así mismo el reconocimiento automático de patrones es muy deseable ya que a través de una clasificación automática se puede detectar e identificar fallos de tal forma que el personal logre tomar una resolución de mantenimiento sin la necesidad de un técnico (Chen et al., 2019).

El presente trabajo tiene como objetivo la comparación de modelos de aprendizaje de máquina para la detección de fallas en turborreactores de doble flujo. Para lo cual se toma en cuenta el mantenimiento basado en la condición de acuerdo al tipo de falla que se muestra y la vida útil de cualquier equipo mediante Machine Learning (Cao et al., 2015). Los resultados obtenidos se comparan con datos reales para verificar la exactitud dando como resultado

el algoritmo Random Forest como el mejor modelo ejecutado con parámetros normales y optimizados con un f1-score de 99.949% y 99.99% respectivamente. Finalmente se conoce que en la base de datos no es posible realizar una extracción de características utilizando aprendizaje automático debido a su peculiaridad en las condiciones operativas.

MATERIALES Y MÉTODOS

En la comparación de los modelos de aprendizaje de máquina para la detección de fallas en turborreactores de doble flujo se utilizó una base de datos proporcionado por el Prognostics Center of Excellence (PCoE) de la NASA, donde se simularon cuatro subconjuntos de datos de entrenamiento y prueba (Devendiran & Manivannan, 2016). En la tabla 1 se muestra los conjuntos de datos con su respectivo número de motores.

El conjunto de datos C-MAPSS contiene las mediciones de 21 sensores instalados en un motor de avión de clase de empuje de 90000 lb, bajo seis condiciones de funcionamiento que van desde alturas al nivel del mar hasta altitudes de 40 000 pies con variaciones de temperatura en el rango de $-51\text{ }^{\circ}\text{C}$ a $39\text{ }^{\circ}\text{C}$ y ángulo de resolución del acelerador (TRA) (20 a 100) (Kopuru et al., 2019).

Tabla 1. Descripción de la base de datos C – MAPSS

Base de datos	C - MAPSS			
	FD001	FD002	FD003	FD004
Entrenamiento	100	260	100	249
Prueba	100	259	100	248
Condiciones operativas	1	6	1	6
Condiciones de falla	1	1	2	2

Fuente: (Kopuru et al., 2019).

Cada fila es una instantánea de los datos tomados durante un solo ciclo operativo, cada columna es una variable diferente; las columnas corresponden al número de unidad tiempo en ciclos, configuración operativa 1-3 y mediciones del sensor 1 al sensor 21, también es importante mencionar que los datos recopilados se encuentran contaminados con el ruido del sensor.

El conjunto de datos de entrenamiento donde la falla crece en magnitud hasta llegar a la falla del sistema, consta de cuatro archivos con su total de datos respectivos como lo muestra la figura 1.

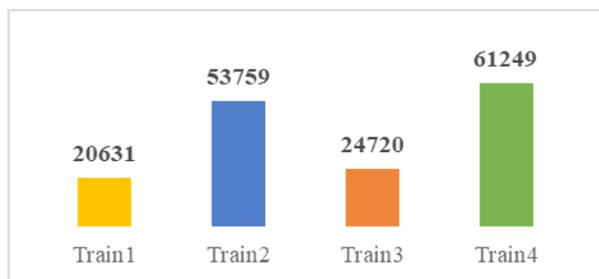


Figura 1. Conjunto de datos de entrenamiento

Fuente: elaboración propia

En conjunto de datos de prueba, su serie temporal finaliza antes que falle el sistema, con el fin de pronosticar el valor total de ciclos funcionales remanentes, es decir, el total de ciclos funcionales después del último ciclo de funcionamiento del motor, se distribuye en cuatro archivos, como lo muestra la figura 2.

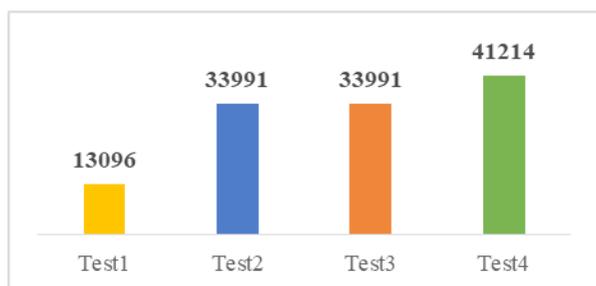


Figura 2. Lectura y comprensión del conjunto de datos

Fuente: elaboración propia

En la figura 3 se puede visualizar el número de datos que contiene cada archivo proporcionó un vector de valores reales de vida útil restante (RUL).

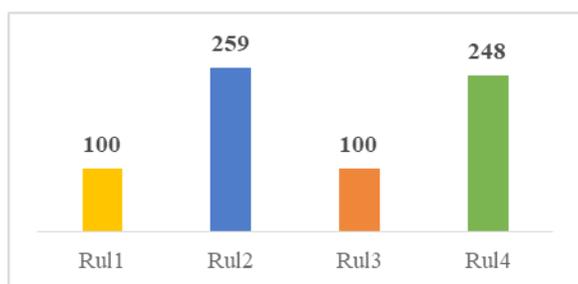


Figura 3. Conjunto de datos RUL

Fuente: elaboración propia

A cada archivo se le asignó un nombre de acuerdo al conjunto de datos que pertenece para posteriormente concatenarlos y formar un nuevo archivo .csv por cada grupo de trabajo, quedando el conjunto de entrenamiento

con 160359 datos, el conjunto de prueba con 104897 datos y el conjunto de datos RUL con 707, como lo muestra la figura 4.

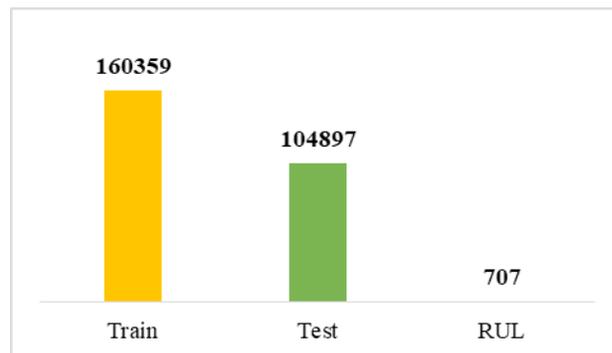


Figura 4. Distribución de datos para cada grupo de trabajo

Fuente: elaboración propia

RESULTADOS Y DISCUSIÓN

De acuerdo con el análisis de la variable objetivo se puede evidenciar un desequilibrio de 141379 datos entre la clase 0 y la clase 1 tal como se ilustra en la figura 5, por tal motivo se aplica el método de sobre-muestreo (oversampling) el cual consisten en crear nuevos datos sintéticos con la finalidad de lograr un equilibrio de datos entre las dos clases.

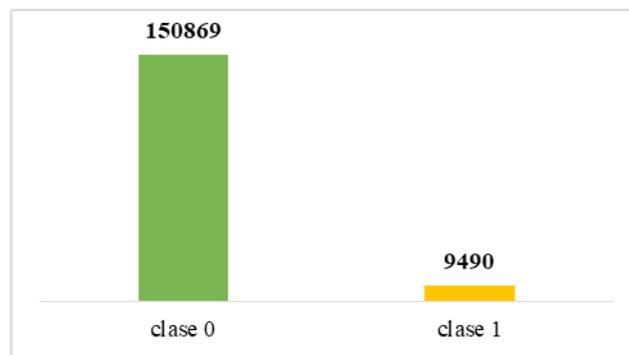


Figura 5. Datos de la variable objetivo del conjunto de entrenamiento

Fuente: elaboración propia

Una vez aplicado el método de sobre-muestreo SMOTE al conjunto de datos, se puede observar un incremento de 138360 datos sintéticos creados para la clase 1 y una reducción considerable en el desequilibrio de datos pasando de 141378 a 3018 datos desequilibrados existentes entre las dos clases de la variable objetivo, como lo indica la figura 6.

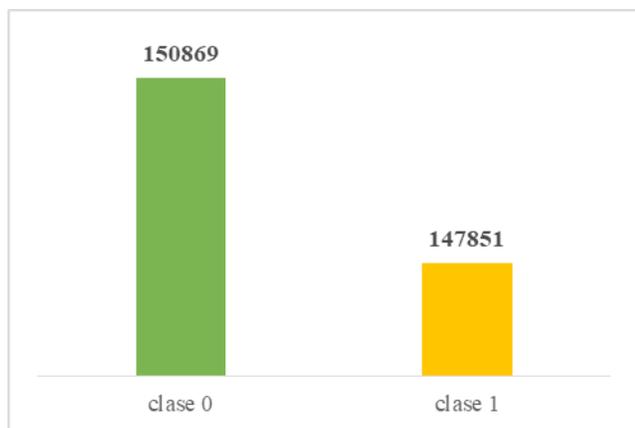


Figura 6. Sobre-muestreo de la variable objetivo

Fuente: elaboración propia

Según el análisis realizado a la variable objetivo denominada clase, se puede evidenciar un desequilibrio de 104283 datos entre la clase 0 y la clase 1, como se expone en la figura 7, por tal motivo se aplica el método de sobre-muestreo (oversampling) con la finalidad de crear nuevos datos sintéticos y lograr un equilibrio de datos entre las dos clases, en correspondencia con los aportes de Ibrahim & Abdulazeez (2021) y Jones et al. (2021).

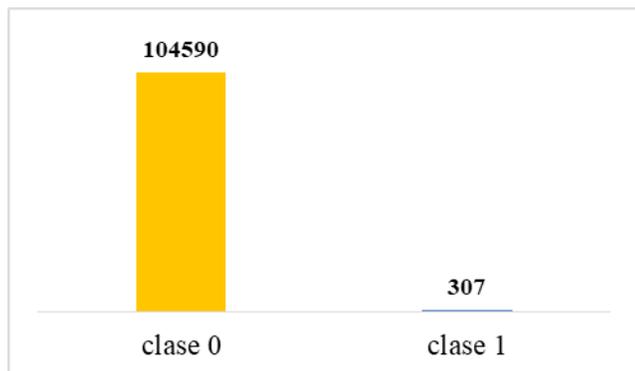


Figura 7. Datos de la variable objetivo del conjunto de prueba

Fuente: elaboración propia

Una vez aplicado el método de sobre-muestreo SMOTE al conjunto de datos, se puede observar un incremento de 102191 datos sintéticos creados para la clase 1 y una reducción considerable en el desequilibrio de datos pasando de 104283 a 2092 datos desequilibrados existentes entre las dos clases de la variable objetivo, tal como lo indica la figura 8.

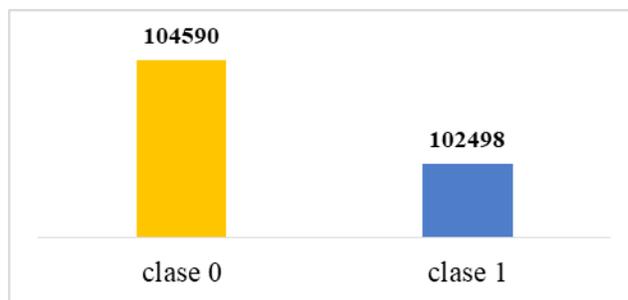


Figura 8. Sobre-muestreo de la variable objetivo

Fuente: elaboración propia

Después de realizar todo el proceso mencionado anteriormente, se puede determinar que el conjunto de datos se encuentra listo para ingresar al clasificador de ML, este procedimiento es muy importante, ya que determinará qué modelo de aprendizaje de máquina es el óptimo para trabajar con el conjunto de datos. Se conoce que existen cinco modelos de ML como es Random Forest, XGBoost, Gradient Boosting y Support Vector Machine; pero para este proceso solo se utilizan cuatro los mismos que se encuentran optimizados y uno sin optimizar.

Con ayuda de la librería scikit-learn y la aplicación de la técnica de validación cruzada se puede obtener la optimización de los parámetros más relevantes, aplicando el método `get_params()`, se puede visualizar una lista de parámetros por defecto del algoritmo siendo estos los más importantes (Probst, 2019):

- `n_estimators`
- `criterion`
- `max_features`
- `max_depth`
- `min_samples_split`
- `min_samples_leaf`
- `bootstrap`

Los parámetros más utilizados por `RandomizedSearchCV` son el número de iteraciones, porque mientras más alto sea este valor mayor será el espacio de búsqueda del clasificador y el método de validación cruzada, el cual reduce las probabilidades de sobreajuste. Se debe tener en cuenta que, si el valor de estos parámetros es un número relativamente alto, el ejecutor tendrá un mayor tiempo de espera. En la tabla 2 se muestran los mejores hiperparámetros que se ajustan al conjunto de datos.

Tabla 2. Mejores hiper parámetros de Random Forest

Nombres	Mejores hiperparámetros
n_estimators	250
criterion	Entropy
min_samples_split	2
max_features	Sqrt
min_samples_leaf	1
max_depth	50
bootstrap	False

A continuación, se ejecutó el método best_estimator para determinar el mejor modelo entrenado, el cual fue empleado para realizar la comprobación del modelo de aprendizaje de máquina. Introducido por primera vez a fines de la década de 1970 por Vapnik, es uno de los algoritmos de aprendizaje basados en kernel, que tiene como objetivo principal resolver un problema de optimización cuadrática convexa para obtener una solución óptima global en teoría y, por lo tanto, superar el dilema extremo local de otras técnicas de aprendizaje automático (Sheykhoua et al., 2020).

El conjunto de datos de entrenamiento se aplicó cinco modelos de aprendizaje de máquina, y a cada modelo se proporcionó una optimización de hiperparámetros con rejilla y se le aplicó el método de validación cruzada de tres iteraciones para estimar el rendimiento de los modelos entrenados y comprobar sus resultados obtenidos en la fase de prueba.

El clasificador Random Forest o bosque aleatorio se puso a prueba con un total de 74,680 muestras distribuidas en 37,749 para la clase 0 es decir, para las muestras sin fallo en los turborreactores y 36,931 para la clase 1, o sea para las muestras que presenta fallos en los turborreactores. Para el total de muestras correspondiente a la clase que no presenta fallos en los turborreactores 36,537 datos fueron clasificados de forma correcta, mientras que 1,212 muestras fueron clasificadas de forma equivocada, como falsos positivos, dando un porcentaje de aciertos de 96,789%. Para el total de muestras obtenidas en la clase que presenta fallos en los turborreactores se identifica que 36880 muestras fueron clasificadas de forma correcta y 51 muestras obtuvieron una clasificación errónea como falsos negativos, obteniendo un porcentaje de 99,862%

Una vez realizado el análisis de los resultados obtenidos de la ejecución del clasificador de bosque aleatorio utilizando parámetros normales y optimizados, se concluye que el porcentaje de aciertos para la clase 0 aumentó en un 0,686%, de igual forma el porcentaje de aciertos para la clase 1 aumentó en un 0,073% y queda comprobado que el clasificador trabaja de mejor forma haciendo una

optimización de hiperparámetros, como lo muestra la figura 9.

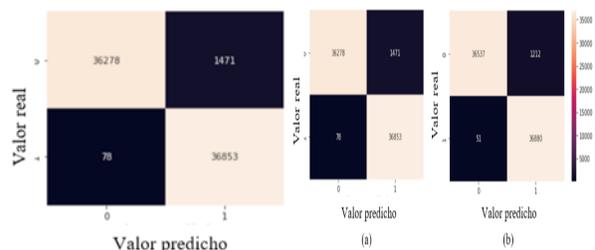


Figura 9. Matriz de confusión del modelo ejecutado con: a) parámetros normales y b) hiperparámetros optimizados

Fuente: elaboración propia

El clasificador XGBoost puso a prueba con un total de 74,680 muestras distribuidas en 37,749 para la clase 0 es decir, para las muestras sin fallo en los turborreactores y 36,931 para la clase 1, es decir para las muestras que presenta fallos en los turborreactores.

Para el total de muestras correspondiente a la clase que no presenta fallos en los turborreactores 36,263 datos fueron clasificados de forma correcta, mientras que 1,486 muestras fueron clasificadas de forma equivocada como falsos positivos, dando un porcentaje de aciertos de 96,063%. Para el total de muestras obtenidas en la clase que presenta fallos en los turborreactores se identifica que 36,726 muestras fueron clasificadas de forma correcta y 205 muestras obtuvieron una clasificación errónea como falsos negativos, obteniendo un porcentaje de 99,445%.

Una vez realizado el análisis de los resultados obtenidos de la ejecución del clasificador de aumento de gradiente utilizando parámetros normales y optimizados, se concluye que el porcentaje de aciertos para la clase 0 aumentó de 1,001%, de igual forma el porcentaje de aciertos para la clase 1 aumentó de 0,393% y queda comprobado que el clasificador trabaja de mejor forma haciendo una optimización de hiperparámetros, como se muestra en la figura 10.

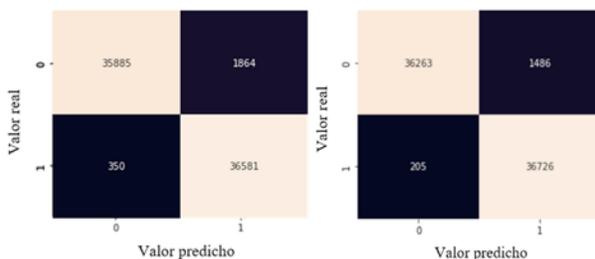


Figura 10 Matriz de confusión del modelo ejecutado con: a) parámetros normales y b) hiperparámetros

Fuente: elaboración propia

optimizados

El clasificador Gradient Boosting se puso a prueba con un total de 74,680 muestras distribuidas en 37,749 para la clase 0 es decir, para las muestras sin fallo en los turborreactores y 36,931 para la clase 1, es decir para las muestras que presenta fallos en los turborreactores.

Para el total de muestras correspondiente a la clase que no presenta fallos en los turborreactores 36,511 datos fueron clasificados de forma correcta, mientras que 1,238 muestras fueron clasificadas de forma equivocada como falsos positivos, dando un porcentaje de aciertos de 96.720%. Para el total de muestras obtenidas en la clase que presenta fallos en los turborreactores se identifica que 36,584 muestras fueron clasificadas de forma correcta y 347 muestras obtuvieron una clasificación errónea como falsos negativos, obteniendo un porcentaje de 99.060%.

Una vez realizado el análisis de los resultados obtenidos de la ejecución del clasificador utilizando parámetros normales y optimizados se concluye que el porcentaje de aciertos para la clase 0 aumentó en un 2.514%, de igual forma el porcentaje de aciertos para la clase 1 aumentó en un 0.411% y queda comprobado que el clasificador trabaja de mejor forma haciendo una optimización de hiperparámetros, como lo muestra la figura 11.

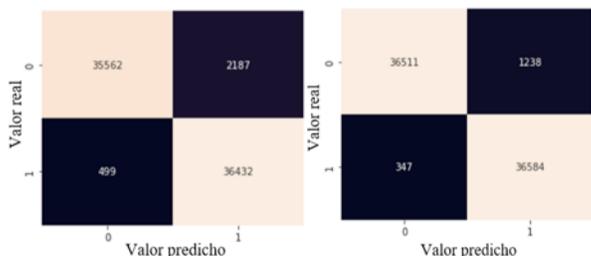


Figura 11. Matriz de confusión del modelo ejecutado con: a) parámetros normales y b) hiperparámetros optimizados

Fuente: elaboración propia

El clasificador XGBoost ejecutado con la optimización de hiperparámetros del Gradient Boosting parámetros por defecto, utilizó un total de 74,680 muestras distribuidas en 37,749 para la clase 0 es decir, para las muestras sin fallo en los turborreactores y 36,931 para la clase 1, correspondiente a las muestras que presentan fallos en los turborreactores. Para el total de muestras correspondiente a la clase que no presenta fallos en los turborreactores, 37,015 datos fueron clasificados de forma correcta, mientras que 734 muestras fueron clasificadas de forma errónea como falsos positivos, dando un porcentaje de aciertos de 98.055%.

Para el total de muestras obtenidas en la clase que presenta fallos en los turborreactores se identificó que 36,823 muestras fueron clasificadas de forma correcta y 108 muestras obtuvieron una clasificación errónea como falsos negativos, obteniendo un porcentaje de 99.707%. En la figura 12 se muestran los resultados obtenidos del modelo entrenado con los parámetros por defecto del bosque aleatorio.

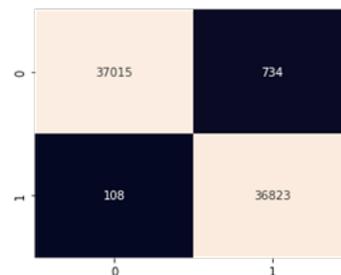


Figura 12. Matriz de confusión del clasificador

Fuente: elaboración propia

El clasificador SVM ejecutado con sus parámetros por defecto utilizó un total de 74,680 muestras distribuidas en 37,749 para la clase 0 es decir, para las muestras sin fallo en los turborreactores y 36,931 para la clase 1, correspondiente a las muestras que presentan fallos en los turborreactores. Para el total de muestras correspondiente a la clase que no presenta fallos en los turborreactores 31,656 datos fueron clasificados de forma correcta, mientras que 6,093 muestras fueron clasificadas de forma errónea como falsos positivos, dando un porcentaje de aciertos de 83.859%.

Para el total de muestras obtenidas en la clase que presenta fallos en los turborreactores se identifica que 34,353 muestras fueron clasificadas de forma correcta y 2,578 muestras obtuvieron una clasificación errónea como falsos negativos, obteniendo un porcentaje de 93.019%. En la figura 13 se muestran los resultados obtenidos del modelo entrenado con los parámetros por defecto del Support Vector Machine.

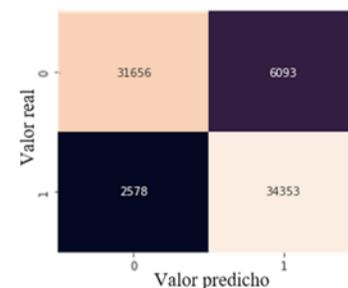


Figura 13. Matriz de confusión clasificador SVM

Fuente: elaboración propia

Al conjunto de datos de prueba se aplicó cinco modelos de aprendizaje de máquina, cada uno fue ejecutado con sus parámetros por defecto y con ayuda de una cuadrícula se aplicó una optimización de hiperparámetros, a excepción del método Support Vector Machine el cual fue entrenado con sus parámetros por defecto.

El clasificador Random Forest o bosque aleatorio se puso a prueba con un total de 51,772 muestras distribuidas en 26,042 para la clase 0 es decir, para las muestras sin fallo en los turborreactores y 25,730 para la clase 1, o sea para las muestras que presentan fallos en los turborreactores. Para el total de muestras correspondiente a la clase que no presenta fallos en los turborreactores 26,028 datos fueron clasificados de forma correcta, mientras que 14 muestras fueron clasificadas de forma equivocada como falsos positivos, dando un porcentaje de aciertos de 99,946%.

Para el total de muestras obtenidas en la clase que presenta fallos en los turborreactores se identifica que 25,722 muestras fueron clasificadas de forma correcta y 8 muestras obtuvieron una clasificación errónea como falsos negativos, obteniendo un porcentaje de 99,969%.

Realizado el análisis de los resultados obtenidos de la ejecución del clasificador de bosque aleatorio utilizando parámetros normales y optimizados, se establece que el porcentaje de aciertos para la clase 0 aumentó en un 0.015%, mientras que el porcentaje de aciertos para la clase positiva se mantiene igual. Esto se sustenta en que los datos del conjunto de prueba no necesitan una optimización de hiperparámetros cuando se ejecuta el modelo de aprendizaje de máquina Random Forest, así como lo muestra la figura 14.

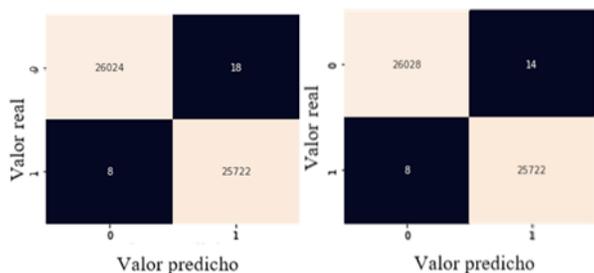


Figura 14. Matriz de confusión del modelo ejecutado con: a) parámetros normales y b) hiperparámetros optimizados

Fuente: elaboración propia

El clasificador XGBoost se puso a prueba con un total de 51,772 muestras distribuidas en 26,042 muestras para la

clase 0 es decir, para las muestras sin fallo en los turborreactores y 25,730 muestras para la clase 1, o sea para las muestras que presenta fallos en los turborreactores.

Para el total de muestras correspondiente a la clase que no presenta fallos en los turborreactores 25,503 datos fueron clasificados de forma correcta, mientras que 539 muestras fueron clasificadas de forma equivocada como falsos positivos, dando un porcentaje de aciertos de 97,930%. Para el total de muestras obtenidas en la clase que presenta fallos en los turborreactores se identifica que 25,675 muestras fueron clasificadas de forma correcta y 55 muestras obtuvieron una clasificación errónea como falsos negativos, obteniendo un porcentaje de 99,786%.

Realizado el análisis de los resultados obtenidos de la ejecución del clasificador de aumento de gradiente extremo utilizando parámetros normales y optimizados, se establece que el porcentaje de aciertos para la clase 0 aumentó en un 6.251%, de igual forma el porcentaje de aciertos para la clase 1 aumentó en un 2,763% y queda comprobado que el clasificador trabaja de mejor forma haciendo una optimización de hiperparámetros, como lo muestra la figura 15.

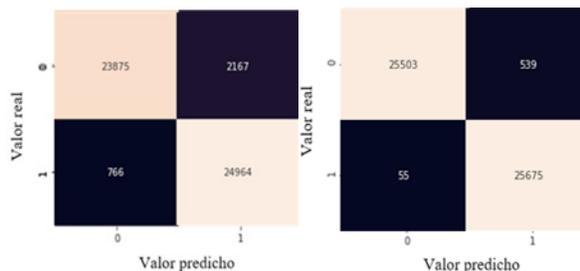


Figura 15. Matriz de confusión del modelo ejecutado con: a) parámetros normales y b) hiperparámetros optimizados

Fuente: elaboración propia

Respecto a la comparación de los modelos de aprendizaje de máquina, la tabla 3, muestra las métricas de evaluación obtenidas del entrenamiento y prueba de los algoritmos de aprendizaje de máquina ejecutados con los parámetros por defecto, donde se puede observar que el modelo de aprendizaje de máquina que se ajusta de mejor forma al conjunto de datos es Random Forest, ya tiene los valores más altos de exactitud, precisión sensibilidad y f1-score, el segundo modelo que mejor se adapta es XGBoost, seguido de Gradient Boosting.

Tabla 3. Métricas de evaluación de los algoritmos ejecutados por defecto

Algoritmo	Exactitud	Precisión	Sensibilidad	F1-score
Random Forest	97.925%	96.162%	99.789%	97.941%
XGBoost	97.035%	95.151%	99.052	95.103%
Gradient Boosting	96.403%	94.337%	98.648%	96.444%
SVM	88.389%	84.935%	93.019%	88.793%

Las métricas de evaluación presentadas en la tabla 4, muestra los algoritmos de aprendizaje de máquina ejecutados con parámetros optimizado, donde se observa que el modelo que mejor se ajusta a la base de datos es el modelo XGBoost optimizado con los hiperparámetros de Gradient Boosting, debido a sus altos porcentajes de precisión, exactitud, f1-score y un área bajo la curva que se aproxima a 100%, en segundo lugar se encuentra Random Forest seguido de Gradient Boosting y en último lugar se encuentra XGBoost.

Tabla 4. Métricas de evaluación de los algoritmos optimizados con hiperparámetros

Algoritmo	Exactitud	Precisión	Sensibilidad	F1-score	AUC
Random Forest	98.308%	96.818%	99.862%	98.316%	99.957%
XGBoost	97.735%	96.111%	99.445%	97.749%	99.280%
Gradient Boosting	97.877%	96.726%	99.060%	98.856%	99.754%
XGBoost optimizado con GB	98.872%	98.045%	99.707%	98.869	99.959%

La tabla 5, muestra las métricas de evaluación obtenidas del entrenamiento y prueba de los algoritmos de aprendizaje de máquina ejecutados con los parámetros por defecto, donde se puede observar que el modelo de aprendizaje de máquina que se ajusta de mejor forma al conjunto de datos es Random Forest ya que tiene los valores más altos de exactitud, precisión sensibilidad y f1-score que mejor se aproxima al 100%, en segundo lugar se encuentra el algoritmo Gradient Boosting seguido de XGBoost.

Tabla 5. Métricas de evaluación de los algoritmos ejecutados con parámetros por defecto

Algoritmo	Exactitud	Precisión	Sensibilidad	F1-score
Random Forest	99.949%	99.930%	99.969%	99.949%
XGBoost	94.335%	92.013%	97.023%	94.452%
Gradient Boosting	99.426%	98.923%	99.934%	99.426%
SVM	83.157%	80.642%	86.992%	83.696%

La tabla 6, muestra los algoritmos ejecutados con parámetros optimizado, donde se observa el modelo que mejor se ajusta a la base de datos es Random Forest debido a sus altos porcentajes de precisión, exactitud, f1-score y un área bajo la curva que se aproxima a 100%, en segundo lugar, se encuentra XGBoost optimizado con los hiperparámetros de Gradient Boosting, seguido de Gradient Boosting y en último lugar se encuentra XGBoost, lo cual se corresponde con el resultados de Wang et al. (2011).

Tabla 6. Métricas de evaluación de los algoritmos ejecutados con optimización de hiperparámetros

Algoritmo	Exactitud	Precisión	Sensibilidad	F1-score	AUC
Random Forest	99.957%	99.945%	99.969%	99.957%	99.999%
XGBoost	98.853%	97.944%	99.786%	98.856%	99.857%
Gradient Boosting	99.867%	99.748%	99.984%	99.866%	99.998%
XGBoost optimizado con GB	99.895%	99.794%	99.996%	99.895%	99.999%

Finalmente, se sintetiza que los resultados obtenidos por los modelos de aprendizaje de máquina desarrollados en cada conjunto de datos señalan al algoritmo Random Forest como el mejor modelo de aprendizaje de máquina que

mejor se adapta a la detección de fallas en turborreactores de doble flujo por tal motivo este modelo puede ser implementado como una alternativa de diagnóstico eficiente, rápida y precisa (Lei et al., 2021) y (Sarajcev et al., 2021).

Para establecer un vínculo en la ciencia, las pruebas estadísticas no son suficientes y se necesitan algunos criterios adicionales para establecer una relación de causalidad con un valor de p-valor inferior a 0.05 con el cuál se rechaza la casualidad de los resultados obtenidos. Por esta razón se utilizó la prueba de permutación para constatar la significancia estadística de la hipótesis planteada al inicio de la investigación, para ello se establecen una hipótesis nula y una hipótesis alternativa.

- H_0 = Utilizando varios métodos de Machine Learning no es posible detectar fallas en turborreactores de doble flujo.
- H_1 = Utilizando varios métodos de Machine Learning es posible detectar fallas en turborreactores de doble flujo.

Después de realizar la comprobación de la hipótesis se obtiene un valor de p-valor igual a 0.01, para todos los modelos de aprendizaje de máquina utilizados en esta investigación los cuales fueron ejecutados en el conjunto de entrenamiento y prueba, a excepción del modelo de aprendizaje de máquina creado a partir de la ejecución del modelo XGBoost el cual está optimizado con los hiperparámetros del modelo Gradient Boosting, el mismo que no arroja un valor de p-valor y a su vez permite generar un bucle en dichos parámetros dando paso a un reinicio cada 4 minutos. El modelo SVM tardó alrededor de 120 horas en ejecutar la prueba de permutación.

CONCLUSIONES

Es posible crear un modelo predictivo automático basado en la lectura de datos tomados desde las señales emitidas por sensores para la detección de fallas en turborreactores de doble flujo mediante Machine Learning. Sin embargo, el modelo de aprendizaje de máquina supervisado con mayor precisión de predicción en el conjunto de datos de entrenamiento y prueba, con parámetros normales e hiperparámetros optimizados de forma correcta es el algoritmo de clasificación Random Forest,

Los resultados obtenidos mediante el entrenamiento y prueba indican que los algoritmos Random Forest, XGBoost y Gradient Boosting generan mejores resultados durante el proceso de comprobación, mientras que el algoritmo SVM genera mejores resultados durante el proceso de entrenamiento, es importante mencionar que el modelo de clasificación SVM no se encuentra optimizado

mediante la utilización de hiperparámetros debido a diversas dificultades generadas al momento de su ejecución.

Los modelos utilizados muestran mejores resultados en sus métricas de evaluación y matriz de confusión después de realizar una optimización de hiperparámetros, por lo tanto, el algoritmo Random Forest se desempeña de mejor forma a pesar que los datos no presentan una extracción de características y no redundan en la predicción debido a su configuración de bosque aleatorio facilitando la predicción de datos de gran dimensión y de estructura centralizada.

Los algoritmos de aprendizaje de máquina presentan mejores resultados en la fase de prueba antes que en la fase de entrenamiento, esto se debe a que la designación de la variable objetivo se basa en los datos RUL generando una mejor afinidad con los datos de la fase de prueba, ya que estos datos son tomados en los últimos ciclos antes que se produzca la falla, mientras que los datos de la fase de entrenamiento sus datos son tomados cuando el motor se encuentra en condiciones estables.

La prueba de hipótesis aplicada a los distintos modelos de aprendizaje de máquina utilizados en la investigación valida la veracidad de los resultados obtenidos no son por casualidad excepto en el modelo de aprendizaje de máquina XGBoost optimizado con los hiperparámetros del modelo Gradient Boosting debido a que se reinicia cada 4 minutos lo que hace imposible realizar una prueba de hipótesis tanto en el conjunto de entrenamiento como en el conjunto de prueba.

REFERENCIAS BIBLIOGRÁFICAS

- Cao, J., Kwong, S., Wang, R., Li, X., Li, K., & Kong, X. (2015). Class-specific soft voting based multiple extreme learning machines ensemble. *Neurocomputing*, *149*, 275-284
- Chen, Z., Gryllias, K., & Li, W. (2019). Mechanical fault diagnosis using convolutional neural networks and extreme learning machine. *Mechanical systems and signal processing*, *133*, 106272.
- Devendiran, S., & Manivannan, K. (2016). Vibration based condition monitoring and fault diagnosis technologies for bearing and gear components-a review. *International Journal of Applied Engineering Research*, *11*(6), 3966-3975.
- Gordón, J. R. (2017). Análisis de prestaciones de un turbofán de tres ejes. *Proyecto fin de carrera Ingeniería Aeronáutica*, Universidad de Sevilla.

- Ibrahim, I., & Abdulazeez, A. (2021). The role of machine learning algorithms for diagnosing diseases. *Journal of Applied Science and Technology Trends*, 2(01), 10-19.
- Jones, I. A., Van Oyen, M. P., Lavieri, M. S., Andrews, C. A., & Stein, J. D. (2021). Predicting rapid progression phases in glaucoma using a soft voting ensemble classifier exploiting Kalman filtering. *Health Care Management Science*, 24(4), 686-701.
- Kopuru, M. S. K., Rahimi, S., & Baghaei, K. T. (2019). Recent approaches in prognostics: State of the art. In *Proceedings on the International Conference on Artificial Intelligence (ICAI)* (pp. 358-365). The Steering Committee of The World Congress in Computer Science, Computer Engineering and Applied Computing (WorldComp).
- Lei, Y., Yang, B., Jiang, X., Jia, F., Li, N., & Nandi, A. K. (2021). Applications of machine learning to machine fault diagnosis: A review and roadmap. *Mechanical Systems and Signal Processing*, 138, 106587.
- Probst, P. (2019). *Hyperparameters, tuning and meta-learning for random forest and other machine learning algorithms* (Doctoral dissertation, Imu).
- Sarajcev, P., Kunac, A., Petrovic, G., & Despalatovic, M. (2021). Power system transient stability assessment using stacked autoencoder and voting ensemble. *Energies*, 14(11), 3148.
- Sheykhmousa, M., Mahdianpari, M., Ghanbari, H., Mohammadimanesh, F., Ghamisi, P., & Homayouni, S. (2020). Support vector machine versus random forest for remote sensing image classification: A meta-analysis and systematic review. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 13, 6308-6325.
- Wang, G., Hao, J., Ma, J., & Jiang, H. (2011). A comparative assessment of ensemble learning for credit scoring. *Expert systems with applications*, 38