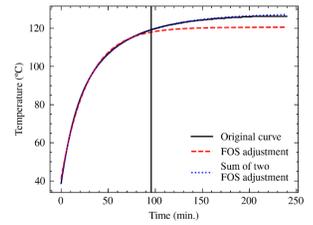


N-BIR: reducción del tiempo del ensayo de BURN-IN de convertidores electrónicos de potencia mediante optimización numérica



N-BIR: a numeric optimization approach for power electronic converter BURN-IN testing time reduction

Iñaki Martínez-Gárriz¹, María-Pilar Herrera-Plaza¹ y Maialen Larretxea-Urrutia²

¹ ASOCIACIÓN INDUSTRIA NAVARRA. Digital Technologies Area. Carretera de Pamplona, 1, 31191 Cordovilla, Navarra, (España).

² INGETEAM POWER TECHNOLOGY. Process and Continuous Improvement. Polígono Teknologi Elkartegia, 48170 Zamudio, Bizkaia, (España).

DOI: <https://doi.org/10.6036/10866> | Recibido: 01/mar/2023 • Inicio Evaluación: 03/mar/2023 • Aceptado: 09/jun/2023

To cite this article: MARTÍNEZ-GÁRRIZ, Iñaki; HERRERA-PLAZA, María-Pilar; LARRETXE-URRUTIA, Maialen. N-BIR: A NUMERIC OPTIMIZATION APPROACH FOR POWER ELECTRONIC CONVERTER BURN-IN TESTING TIME REDUCTION. *DYNA*. March - Abril 2024. Vol.99, n.2, pp. 201-207. DOI: <https://doi.org/10.6036/10866>

FINANCIACIÓN

Este trabajo ha sido apoyado y financiado por el Gobierno de Navarra (GdN) y el Fondo Europeo de Desarrollo Regional (FEDER) en la convocatoria de ayudas para la realización de proyectos de I+D en 2021.

ABSTRACT

- Burn-in testing is an effective method for detecting early faults in electronic products before they reach the market. This test has a high cost due to lengthy test time on a test bench. In this paper, we propose N-BIR (Numeric optimization approach for power electronic converter Burn-In testing time Reduction), an algorithm capable of predicting the burn-in test temperature of power electronic converters, intending to shorten the duration of such tests. This algorithm optimizes by least squares a theoretical model of the system, using as data a fraction of the total burn-in test. Moreover, not only is it capable of making accurate predictions, but it also accompanies them with a prediction interval, so that the algorithm itself can quantify how confident it is of its predictions. We show that using 40% of a conventional rolling test total, the proposed algorithm outperforms several of today's most common Machine Learning algorithms. Furthermore, we show that it can reduce burning time by 50% to 60% by making accurate predictions, which makes it possible to identify a significant portion of converters that don't require full testing, ultimately lowering costs and boosting productivity.
- **Keywords:** Burn-in temperature prediction, Burn-in time reduction, Levelized Cost of Energy, Machine Learning, Electronic Power Converter, Reliability.

RESUMEN

El test de burn-in es un método efectivo para detectar fallos tempranos en productos electrónicos antes de que éstos lleguen al mercado. Este test tiene un alto coste por la duración de las pruebas en un banco de pruebas. En este trabajo proponemos N-BIR (Numeric optimization approach for power electronic Burn-In test time Reduction), un algoritmo capaz de predecir la temperatura del test de burn-in de convertidores electrónicos de potencia, con la intención de acortar la duración de dichas pruebas. Este algoritmo optimiza por mínimos cuadrados un modelo teórico del sistema, utilizando como datos una fracción del ensayo total de

burn-in. Además, no sólo es capaz de realizar predicciones precisas, sino que proporciona un intervalo de predicción, de modo que el propio algoritmo puede cuantificar el grado de seguridad de sus predicciones. Demostramos que utilizando el 40% del total de un test burn-in, el algoritmo propuesto supera a varios de los algoritmos de Machine Learning más habituales en la actualidad. Además, demostramos que puede reducir el tiempo de combustión entre un 50% y un 60% realizando predicciones precisas lo que permite identificar una gran parte de convertidores que no requieren llevar a cabo el test completo, lo que en última instancia reduce los costos y aumenta la productividad.

Palabras Clave: Predicción de temperatura de burn-in, reducción del tiempo de burn-in, Coste de energía nivelado, Machine Learning, convertidor electrónico de potencia, Fiabilidad.

1. INTRODUCCIÓN

A pesar de la ralentización económica provocada por la pandemia de COVID-19, la capacidad de las energías renovables aumentó en 257 GW en 2021, según la Agencia Internacional de Energías Renovables (IRENA¹). Esto contrasta con el declive experimentado por otras partes del sector energético, como el petróleo, el gas y el carbón. La capacidad de la energía solar fotovoltaica (FV) aumentó en 145 GW en 2021, y la Agencia Internacional de la Energía (AIE²) predijo que seguiría creciendo de forma constante, hasta alcanzar casi 161 GW en 2022, acelerándose a lo largo de 2023 y 2025. La capacidad de energía eólica también experimentó un crecimiento, y el Consejo Mundial de la Energía Eólica (GWEC³) informó de un aumento de 94 GW en 2021, y de un crecimiento previsto de aproximadamente 110 GW en 2022 y 2023, y de aproximadamente 130 GW en 2025.

Para hacer frente a la rápida expansión mundial y al aumento de la capacidad instalada de los sistemas de energía renovable, hay que afrontar grandes retos para evolucionar y reinventar la tecnología con el fin de mejorar su competitividad frente al resto. Estos retos incluyen la reducción del coste nivelado de la energía (Levelized Cost of Energy, LCoE) de las centrales mediante estra-

¹ <https://www.irena.org>

² <https://www.iea.org/>

³ <https://gwec.net/>

tegias como la reducción de los costes de los componentes (entre otras).

En este contexto, la reducción de los costes de fabricación de los convertidores electrónicos de potencia es crucial. Actualmente, el test de burn-in (BI) es el más costoso del proceso de fabricación en términos de tiempo y dinero, debido a la duración del test y al uso de bancos de pruebas que requieren una elevada inversión. Este test de BI es esencial ya que detecta fallos de ensamblaje e identifica desviaciones en componentes desarrollados bajo especificación. También ayuda a identificar cambios en el proceso de fabricación de algunos componentes o materias primas que pueden afectar a la calidad y la vida útil del producto, evitando así la entrega de productos que no cumplan los estándares de alta calidad. En la actualidad, el porcentaje de convertidores de potencia que no superan el test de BI oscila entre el 5 y el 7% dependiendo del modelo de convertidor. El test a consiste en hacer funcionar al convertidor de potencia a su potencia nominal mientras se mide en tiempo real la temperatura en todas las zonas críticas. Si alguna temperatura supera un valor límite después de la estabilización, la prueba falla y el convertidor debe ser revisado. En caso contrario, la prueba es satisfactoria. Acortar la duración de esta prueba aumentaría la capacidad de producción y reduciría los costes de fabricación.

Muchos estudios han intentado disminuir la duración de este test, no sólo en el campo de la electrónica de potencia, sino en todos los campos relacionados con la electrónica. Daasch *et al.* proponen en [1] el uso de pruebas estadísticas multivariantes. Uno de los trabajos pioneros en este campo es [2], en el que los autores realizan una reducción dimensional de los datos mediante el análisis de componentes principales (PCA) para posteriormente realizar las pruebas. Así, demostraron que el PCA puede identificar las piezas más susceptibles de fallar en el futuro. En [3], Makris *et al.* demuestran cómo las pruebas multivariantes también son eficaces para detectar futuros fallos en dispositivos de radiofrecuencia (RF). En [4], los autores emplean de nuevo pruebas paramétricas multivariantes para la reducción del tiempo de BI, donde demuestran que 10h de una prueba BI de 48h son suficientes para detectar la mayoría de los fallos, acortando así las pruebas en gran medida.

En los últimos años, se han empezado a utilizar técnicas de aprendizaje automático (ML) e inteligencia artificial (AI) para abordar los problemas mencionados. En [5], los autores emplean una máquina de vectores de soporte (SVM) para predecir el fallo prematuro de los láseres cuánticos en cascada durante sus respectivas pruebas de BI, con lo que pueden detectar fallos 200h antes, que utilizando únicamente la prueba de BI. En [6], los autores modelan la consecuente degradación del test BI usando procesos de Wiener transformados en el tiempo que optimizan usando algoritmos evolutivos multi-objetivo. Por otro lado, en [7] los autores emplean redes neuronales convolucionales (CNN) para detectar fallos tempranos y en [8] los autores adoptan un enfoque denominado estudio de burn-in, que se centra en extraer una muestra de dispositivos de la producción en marcha y estudiarlos.

Este estudio pretende desarrollar un algoritmo capaz de predecir la temperatura final de BI antes de que el convertidor la alcance para acortar el tiempo necesario para realizar la prueba. De esta forma, un convertidor podrá considerarse bueno o malo antes de que finalice su prueba de BI, lo que supondrá un aumento de la productividad y un ahorro económico en el proceso productivo y, en consecuencia, una mejora del LCoE. Además, otro de los objetivos del estudio es determinar cuánto se puede acortar la duración del test de BI garantizando los niveles de calidad requeridos.

El artículo se estructura como sigue: en primer lugar, se describe el funcionamiento del algoritmo propuesto. A continuación, se detallan la metodología y los materiales utilizados en los experimentos. Posteriormente, se presentan los resultados obtenidos y su correspondiente discusión. Por último, se presentan las conclusiones generales del estudio, así como posibles líneas de investigación futuras.

2. ALGORITMO PROPUESTO: N-BIR

2.1. PREDICCIÓN DE LA TEMPERATURA FINAL

Durante el test de BI, la curva de calentamiento del convertidor de potencia puede variar en función del diseño específico del dispositivo y de las condiciones de la prueba. En los tres modelos de convertidor de potencia utilizados en este trabajo, el comportamiento de las curvas de temperatura puede aproximarse a un sistema de primer orden (FOS), que se muestra en (1):

$$T(t) = K \left(1 - e^{-\frac{t}{\tau}} \right) + \phi \quad (1)$$

donde:

- K representa el aumento de temperatura a lo largo del ensayo.
- ϕ representa la temperatura inicial del conductor.
- τ representa el tiempo de estabilización del calentamiento.

El método propuesto consiste en ajustar los valores de τ , ϕ y K a partir de las medidas iniciales observadas, para predecir a qué temperatura se estabilizará la curva por un ajuste por mínimos cuadrados. Este método es un método de optimización numérica que consiste en encontrar los valores óptimos de los parámetros de una función, tales que minimicen los cuadrados de los residuos entre dicha función y una serie de valores numéricos. Es decir, trata de encontrar los parámetros de la función que más se aproximen a dichos valores numéricos.

Aunque inicialmente se ha utilizado (1) para ajustar la curva, dado que en algunas pruebas largas se han observado tendencias ligeramente ascendentes cuando el sistema ya debería haber alcanzado su estado estacionario, en tales casos, hemos utilizado la suma de dos FOS mostrada en (2), que es una aproximación a un cambio en uno de los flujos de calor durante la prueba (ya sea generación interna por corriente eléctrica o enfriamiento convectivo).

$$T(t) = K_1 \left(1 - e^{-\frac{t}{\tau_1}} \right) + K_2 \left(1 - e^{-\frac{t}{\tau_2}} \right) + \phi \quad (2)$$

La Fig. 1 muestra un ejemplo de esta casuística: la curva azul discontinua (curva ajustada utilizando (2)) se ajusta a la curva real (curva negra) mucho mejor que la curva roja discontinua (curva ajustada utilizando (1)) cuando t tiende a ser alto.

El algoritmo elige entre (1) y (2) empezando siempre por (1) y observando su tiempo de estabilización en el 98,2% (4τ). Siendo t_{test} la duración teórica total de la prueba: si el tiempo de estabilización es pequeño comparado con t_{test} (menos del 60% de t_{test}), se considera que el sistema se estabiliza demasiado rápido y, por tanto, utiliza (2). En caso contrario, toma (1) como válido. De este modo, si el FOS encontrado se estabiliza demasiado pronto para la duración total de la prueba, utilizamos la suma de dos FOS para remediarlo. Sin embargo, el τ de la segunda FOS está acotada de forma que sea mucho más lenta que la primera, para que sólo mo-

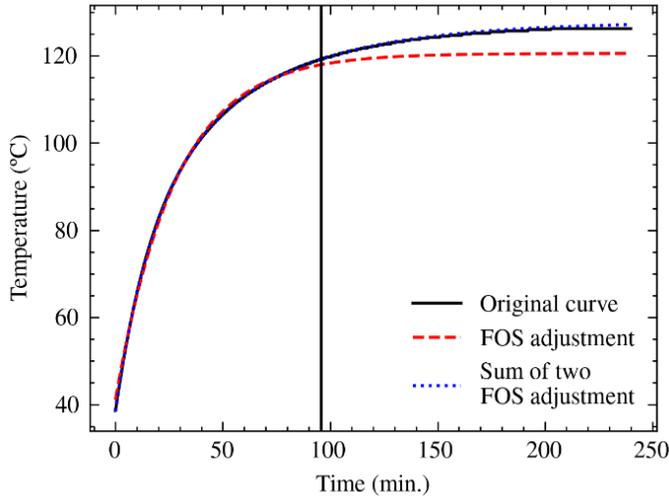


Fig. 1 Ejemplo de ajuste de un FOS y una suma de dos FOS a la misma curva. Los puntos utilizados para ajustar ambas curvas son los situados a la izquierda de la línea vertical.

dele estas tendencias finales al alza, pero no modifique demasiado la dinámica del sistema. En concreto, se ha acotado entre 450 y 1950. En cuanto a la inicialización de los parámetros, todos se han inicializado a partir de las medias observadas en el conjunto de datos, excepto ϕ que se ha inicializado con el valor de temperatura inicial observado, ya que éste es conocido. La inicialización de cada parámetro antes de los valores de optimización para (1) y (2) son $K = K_1 = 70$, $\tau = \tau_1 = 1500$, $\phi = T[0]$, $K_2 = 0$ y $\tau_2 = 500$.

2.2. GENERACIÓN DEL INTERVALO DE PREDICCIÓN

El algoritmo no sólo predice la temperatura final de BI, sino que también proporciona un intervalo de predicción (PI), es decir, dos valores de temperatura que limitan la temperatura final. De este modo, el usuario puede ver el grado de confianza del algoritmo en su predicción.

```

Algorithm 1: N-BIR algorithm
Data:  $T_n$  measured and filtered temperature values,  $t_{current}$  current test time,  $t_{test}$  total duration of the test,  $\Omega$  bounds of parameters to optimize
Result:  $T_{predicted}^f$ ;  $T_{lower\ bound}^f$ ;  $T_{upper\ bound}^f$ 
 $K^f, \tau^f, \varphi^f = \text{AdjustFOS}(T_n)$ ;
 $t_s = A\tau^f$ ;
if  $t_s \geq 0,6 \cdot t_{current}$  then
    /* FOS works */
     $\theta^f = [K^f, \tau^f, \varphi^f]$ ;
     $f = f_{FOS}$ ;
else
    /* FOS do not work, use sum of two FOS instead */
     $K_1^f, \tau_1^f, K_2^f, \tau_2^f, \varphi^f = \text{AdjustSumOf2FOS}(T_n)$ ;
     $\theta^f = [K_1^f, \tau_1^f, K_2^f, \tau_2^f, \varphi^f]$ ;
     $f = f_{FOS1} + f_{FOS2}$ ;
end
 $T_{predicted}^f = f(\theta^f)(t_{test})$ ;
 $T_{lower\ bound}^f, T_{upper\ bound}^f = T_{predicted}^f$ ;
 $\epsilon = \text{MSE}(T_n, f(\theta^f))$ ;
/* Prediction band */
for  $\theta_{fixed}^f \in \theta^f$  do
     $\theta^f = \theta_{fixed}^f \cap \theta^f$ ;
    for  $i \leftarrow 0$  to 49 por 1 do
         $\rho = \theta_{fixed}^f(0,75 + 0,01 \cdot i)$ ;
         $\theta^f = \text{AdjustWithFixedParam}(T_n, \rho)$ ;
         $\hat{\epsilon} = \text{MSE}(T_n, f(\theta^f))$ ;
        if  $\epsilon - \hat{\epsilon} \leq 0,1$  then
             $T_{fixed\ param.} = f(\theta^f, \rho)(t_{test})$ ;
            if  $T_{fixed\ param.} \leq T_{lower\ bound}^f$  then
                 $T_{lower\ bound}^f = T_{fixed\ param.}$ ;
            end
            if  $T_{fixed\ param.} \geq T_{upper\ bound}^f$  then
                 $T_{upper\ bound}^f = T_{fixed\ param.}$ ;
            end
        end
    end
end
Return  $T_{predicted}^f$ ;  $T_{lower\ bound}^f$ ;  $T_{upper\ bound}^f$ ;
    
```

Fig. 2 Pseudocódigo de N-BIR.

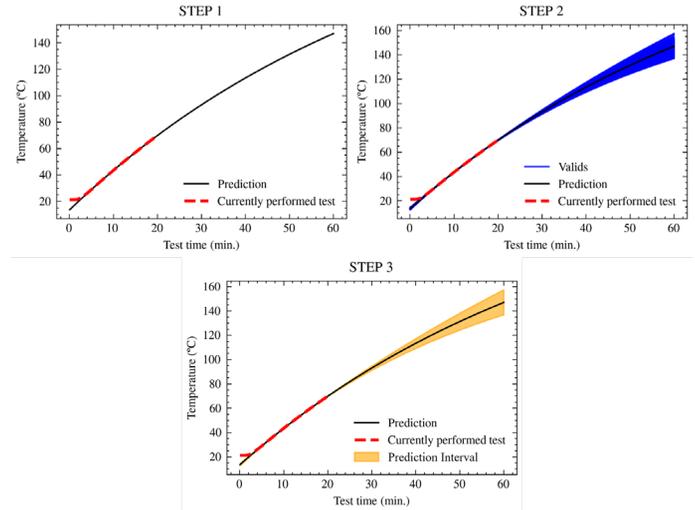


Fig. 3 Visualización paso a paso del algoritmo propuesto.

Para ello, para cada parámetro de la curva ajustada, el algoritmo calcula 50 valores equidistantes desde ese valor menos su 25% hasta ese valor más su 25%. Para cada uno de estos valores, el algoritmo vuelve a optimizar la curva a las muestras manteniendo fijo el valor del parámetro. Si el resultado es similar al original sin restricciones (con un RMSE como máximo un 10% superior al original), se almacena como satisfactorio. Una vez completado este proceso para todos los valores de los parámetros, el algoritmo devuelve las temperaturas finales más alta y más baja de entre todas las curvas satisfactorias como intervalo de predicción.

El proceso completo de predicción N-BIR se muestra en la Fig. 2:

La Fig. 3 representa gráficamente las etapas mencionadas. En la etapa 1, el algoritmo predice la curva de temperatura a partir de los valores medidos hasta ese momento. A continuación, en la etapa 2, genera todas las curvas posibles basándose en el muestreo descrito anteriormente. Por último, en la etapa 3, utiliza las envolventes entre estas curvas para crear el intervalo de predicción. El ejemplo de la figura consiste en predecir la temperatura de una prueba de 60 minutos utilizando sólo los primeros 20 minutos de datos.

3. MATERIALES Y MÉTODOS

3.1. MATERIALES: DATOS

Los datos utilizados provienen de tres modelos diferentes de convertidores de potencia back-to-back de baja tensión, de modelos eólicos terrestres y marinos, y fotovoltaicos, con potencias de 4 a 8 MW. Por razones de confidencialidad, no es posible facilitar más detalles sobre las características técnicas de los productos. Durante el test de BI, la temperatura se midió simultáneamente en varios puntos de cada convertidor. Se utilizaron datos de un total de 154 convertidores y 850 curvas de temperatura. En concreto, había 60 convertidores del Modelo 1 con 2 curvas de temperatura por convertidor y una duración de la prueba de 1 hora; 44 convertidores del Modelo 2 con 3 curvas de temperatura por convertidor y una duración de la prueba de 1 hora; y, por último, 50 convertidores del Modelo 3 con 12 curvas por convertidor y una duración de la prueba de 4 horas.

Dado que el algoritmo propuesto utiliza el comportamiento inicial de la curva de temperatura, no sólo es interesante que pueda realizar predicciones precisas, sino también que pueda hacerlo utilizando el menor número posible de datos iniciales. Así, cuantos

menos valores de temperatura se necesiten para generar predicciones precisas, mayor será el potencial para acortar la duración del test de BI.

Para evaluar el algoritmo en función del porcentaje del test total utilizado para la predicción y estimar el potencial de reducción de la duración de la prueba, generamos múltiples instancias a partir de cada curva. Estas instancias consisten en rangos de porcentajes iniciales de la curva total, que van del 15% al 90% en incrementos del 5%. El resultado es un total de 13.600 instancias (850 curvas para cada porcentaje). Para eliminar el ruido de las muestras, filtramos todas las instancias utilizando el filtro Savitzky-Golay [9] con un tamaño de ventana del 5,2% del número total de puntos y un orden polinómico de 3.

3.2. MÉTODOS

3.2.1. Metodología de evaluación

Para evaluar el rendimiento de nuestro algoritmo, lo comparamos con los cinco algoritmos más avanzados de aprendizaje automático:

- **Regresión logística:** pertenece al grupo de los clasificadores lineales. Su objetivo es modelizar la relación entre un conjunto de características de entrada (variables independientes) y una variable de resultado binaria (variable dependiente). Supone que la relación entre las características y las probabilidades logarítmicas de la variable de resultado puede modelarse mediante una función lineal.
- **Regresión K-Nearest Neighbour:** técnica de aprendizaje por instancias o regresión no paramétrica que predice el valor de una variable objetivo basándose en los valores de sus vecinos más cercanos. El algoritmo consiste en encontrar el número K de ejemplos de entrenamiento más cercanos a los datos de entrada y, a continuación, calcular la media de los resultados de esos ejemplos de entrenamiento para hacer una predicción.
- **Random Forest [10, 11]:** método de predicción basado en un conjunto de árboles de decisión (DT) entrenados mediante el método bagging, que mejora la precisión del predictor combinando las salidas de múltiples modelos.
- **eXtreme Gradient Boosting [12]:** XGBoost es un algoritmo de boosting basado en DT y en el método Gradient Boosting [13]. Entre otras cosas, difiere de este método en el uso de un término de regularización en su función de costes.
- **CatBoost [14]:** algoritmo de boosting que funciona especialmente bien en conjuntos de datos que contienen variables categóricas. Sin embargo, también da buenos resultados con variables numéricas. Para transformar variables categóricas en numéricas, en lugar de utilizar la codificación One-Hot como en la mayoría de algoritmos, CatBoost utiliza una codificación denominada Target-based encoding, que consiste en asignar un valor numérico a cada una de las categorías en función de la variable objetivo.

Para entrenar modelos ML, es necesario separar el conjunto de datos en conjuntos de entrenamiento y de test. Dado que ambos conjuntos deben contener ejemplos de todas las duraciones del test de BI (del 15% al 90% en incrementos del 5%), en lugar de hacer la separación en función de las instancias, se ha hecho en función de los convertidores. Además, como los 3 modelos de convertidor tienen sus particularidades, se ha hecho una separación estratificada por modelo de convertidor. De esta forma, todos los modelos están igualmente representados en ambos conjuntos. Finalmente hemos utilizado aproximadamente el 23% de los convertidores para testing y el resto para

entrenamiento, lo que ha dado como resultado 10480 instancias de entrenamiento y 3120 instancias de test. Por tanto, de las 850 curvas disponibles, se han obtenido 13600 instancias (850 por cada uno de los porcentajes). Cada modelo, Modelo1, Modelo2 y Modelo3, tiene 1472, 1536 y 747 instancias para el entrenamiento y 432, 576 y 2112 para test, respectivamente.

Para entrenar los modelos ML, se han extraído las siguientes características de entrada de cada instancia:

- **Tiempo de prueba:** tiempo que ha durado la prueba antes de intentar predecir su valor de temperatura final. Es decir, los minutos transcurridos para obtener las mediciones utilizadas para generar la instancia.
- **Gradiente de temperatura:** gradiente medio de temperatura que se ha producido a lo largo de las mediciones (3):

$$\frac{T_{\text{last measured}} - T_0}{\Delta t} \quad (3)$$

- **Gradiente máximo:** el gradiente máximo observado en las temperaturas medidas.
- **Instante de tiempo del gradiente máximo:** instante de tiempo en la que se observa el gradiente máximo.

Así, de cada curva se han obtenido cuatro variables numéricas, suficientes para caracterizar un FOS. Como el valor inicial de la temperatura no se ha introducido en el algoritmo, la variable dependiente que debe predecirse pasa a ser el incremento de temperatura (ΔT) durante el ensayo, de modo que $T_{\text{FINAL}} = T_0 + \Delta T$.

Dado que el algoritmo propuesto está basado en la optimización numérica (no aprende de datos pasados, simplemente utiliza los datos de la instancia a predecir), no requiere los conjuntos de entrenamiento y prueba. Sin embargo, para realizar una comparación lo más rigurosa posible entre los modelos ML y el algoritmo propuesto, este último sólo se ha evaluado sobre las instancias pertenecientes al conjunto de prueba, ya que son los puntos sobre los que se han evaluado el resto de modelos de ML.

3.2.2. Métricas de evaluación

Hemos dividido la evaluación del algoritmo propuesto en dos partes. En primer lugar, una comparación de su capacidad de regresión respecto a algoritmos ML entrenados y, en segundo lugar, un análisis del intervalo de predicción generado. Respecto a la primera parte, dado que se trata de un problema de regresión, se han utilizado las siguientes métricas comúnmente aplicadas en este tipo de problemas:

- **Error absoluto medio (MAE, 4):** Calcula la media del valor absoluto de los residuos (diferencias entre el valor previsto y el real).

$$MAE = \frac{\sum_{i=1}^n |\hat{y}_i - y_i|}{n} \quad (4)$$

- **Error cuadrático medio (RMSE, 5):** Calcula la raíz media de los residuos al cuadrado.

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{n}} \quad (5)$$

- **Coefficiente de determinación (R^2 , 6):** El coeficiente de determinación es la proporción de la varianza total de la variable que explica la regresión. También se denomina R^2 y sirve para reflejar la bondad del ajuste de un modelo a la variable que se quiere explicar. R^2 oscila normalmente entre 0 y 1, pero hay casos en los que puede arrojar valores negativos [15].

$$R^2 = 1 - \frac{\sum_{i=1}^n (\hat{y}_i - y_i)^2}{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2} \quad (6)$$

donde:

- \hat{y}_i representa la predicción realizada por el algoritmo para la i-ésima instancia.
- y_i representa el valor real de la temperatura de BI.
- n representa el número de instancias de prueba.

Para la segunda parte del análisis de resultados, el análisis PI, hemos utilizado las siguientes métricas [16]:

- **Probabilidad de cobertura del intervalo de predicción (PI):** Media de los valores binarios, 1 si la temperatura final real está dentro de PI, 0 en caso contrario.
- **Anchura media del intervalo de predicción (MPIW):** media de la diferencia entre los límites superior e inferior del PI.

4. RESULTADOS

4.1. PREDICCIÓN DE LA TEMPERATURA BI FINAL

Según los resultados presentados en Tabla 1 y en la Fig. 4, LR es el algoritmo con peores resultados, resultado esperable ya que es el más sencillo de todos. Los algoritmos basados en árboles de decisión ofrecieron mejores resultados, siendo CatBoost significativamente mejor que RF y XGBoost, y el algoritmo ML con mejor rendimiento. KNN, por su parte, ofreció una precisión ligeramente inferior a CatBoost.

El algoritmo propuesto, N-BIR, superó a todos los modelos ML, siempre que se utilizara una fracción de la curva igual o superior al 40%. Esto se debe a que, al tratarse de un modelo basado en la optimización numérica, se ajusta sólo en función de la instancia que quiere predecir, por lo que necesita más información de la instancia actual que los otros modelos ML que han sido entrenados en todo el conjunto de entrenamiento. Sin embargo, utilizar sólo

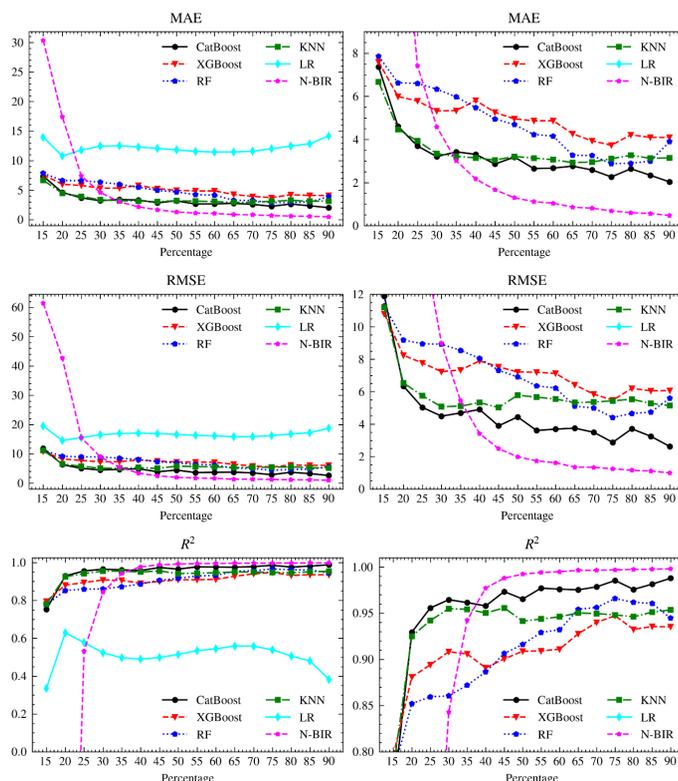


Fig. 4 Valores MAE, RMSE y R2 obtenidos en el conjunto de pruebas por cada algoritmo frente al porcentaje de la prueba total utilizado para predecir. La figura de la derecha es una ampliación de la figura de la izquierda.

el 40% de la curva significa acortar el tiempo de prueba de BI en un 60%, lo que es un resultado muy bueno.

Este fenómeno se observa más claramente en la Tabla 2 en la que hemos realizado la prueba U de Mann-Whitney (la prueba U de Mann-Whitney es la versión no paramétrica de la prueba t de Student) sobre los residuos absolutos obtenidos por N-BIR frente a los obtenidos por los otros algoritmos (de nuevo por porcentajes).

Percentage	MAE						RMSE						R ²					
	CatBoost	XGBoost	RF	KNN	LR	N-BIR	CatBoost	XGBoost	RF	KNN	LR	N-BIR	CatBoost	XGBoost	RF	KNN	LR	N-BIR
15%	7,360	7,591	7,855	6,670	13,958	30,341	11,895	10,792	11,281	11,191	19,482	61,419	0,752	0,796	0,777	0,781	0,335	-6,320
20%	4,602	5,990	6,626	4,471	10,789	17,415	6,348	8,244	9,183	6,540	14,553	42,645	0,929	0,881	0,852	0,925	0,629	-2,529
25%	3,696	5,784	6,602	3,931	11,781	7,412	5,030	7,770	8,954	5,750	15,534	15,527	0,956	0,894	0,859	0,942	0,577	0,532
30%	3,202	5,326	6,332	3,340	12,450	4,579	4,495	7,238	8,926	5,074	16,503	8,992	0,965	0,908	0,861	0,955	0,523	0,843
35%	3,424	5,338	5,974	3,213	12,519	3,016	4,689	7,337	8,550	5,112	16,959	5,442	0,962	0,906	0,872	0,954	0,497	0,943
40%	3,309	5,810	5,479	3,145	12,311	2,164	4,905	7,899	8,057	5,328	17,077	3,423	0,958	0,891	0,886	0,950	0,490	0,977
45%	2,859	5,254	4,950	3,043	12,059	1,667	3,895	7,535	7,312	5,027	16,930	2,492	0,973	0,901	0,906	0,956	0,498	0,988
50%	3,183	4,955	4,688	3,209	11,807	1,295	4,447	7,216	6,914	5,782	16,648	1,989	0,965	0,909	0,916	0,941	0,515	0,992
55%	2,648	4,864	4,229	3,133	11,562	1,114	3,611	7,198	6,359	5,666	16,305	1,739	0,977	0,909	0,929	0,944	0,534	0,994
60%	2,671	4,869	4,156	3,060	11,448	1,041	3,695	7,125	6,224	5,547	16,140	1,620	0,976	0,911	0,932	0,946	0,544	0,995
65%	2,761	4,263	3,271	2,926	11,446	0,856	3,754	6,425	5,111	5,325	15,857	1,355	0,975	0,928	0,954	0,950	0,559	0,996
70%	2,585	3,933	3,249	2,954	11,580	0,810	3,501	5,848	4,995	5,362	15,887	1,338	0,979	0,940	0,956	0,950	0,558	0,997
75%	2,260	3,734	2,875	3,106	11,995	0,685	2,880	5,482	4,411	5,437	16,210	1,254	0,985	0,947	0,966	0,948	0,539	0,997
80%	2,640	4,214	2,892	3,265	12,464	0,599	3,715	6,201	4,663	5,538	16,778	1,156	0,976	0,933	0,962	0,946	0,506	0,997
85%	2,340	4,097	2,944	3,126	12,854	0,562	3,251	6,058	4,740	5,276	17,211	1,111	0,981	0,936	0,961	0,951	0,480	0,998
90%	2,028	4,093	3,894	3,144	14,179	0,473	2,624	6,069	5,606	5,142	18,743	0,998	0,988	0,935	0,945	0,954	0,383	0,998

Tabla 1 Métricas de evaluación obtenidas por cada algoritmo en el conjunto de pruebas para cada una de las fracciones de la prueba total utilizada.

Percentage	CatBoost		XGBoost		N-BIR vs RF		KNN		LR	
	p-val	Best	p-val	Best	p-val	Best	p-val	Best	p-val	Best
15%	$< 10^{-3}$	CatBoost	$< 10^{-3}$	XGBoost	$< 10^{-3}$	RF	$< 10^{-3}$	KNN	0.001	LR
20%	$< 10^{-3}$	CatBoost	$< 10^{-3}$	XGBoost	0.002	RF	$< 10^{-3}$	KNN	0.316	=
25%	0.007	CatBoost	0.481	=	0.077	=	0.003	KNN	$< 10^{-3}$	N-BIR
30%	0.388	=	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	0.358	=	$< 10^{-3}$	N-BIR
35%	0.001	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	0.142	=	$< 10^{-3}$	N-BIR
40%	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR
45%	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR
50%	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR
55%	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR
60%	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR
65%	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR
70%	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR
75%	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR
80%	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR
85%	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR
90%	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR	$< 10^{-3}$	N-BIR

Tabla 2 Pruebas U de Mann-Whitney ($\alpha=0,05$) entre los residuos absolutos obtenidos en el conjunto de prueba con N-BIR en comparación con los demás algoritmos utilizados.

Así, si no se rechaza la hipótesis nula (H0), significa que ambos residuos pertenecen a la misma población, es decir, que ambos algoritmos cometen el mismo error y, en consecuencia, tienen un rendimiento similar. Por el contrario, si se rechaza la hipótesis nula, se cumplirá la hipótesis alternativa (H1) de que ambos residuos pertenecen a poblaciones diferentes, lo que indica que uno de los algoritmos comete un error menor que el otro. Dado que la prueba compara las medianas de las distribuciones, tanto los valores muy altos como los muy bajos de los residuos en este caso pueden dar lugar a una mediana similar si son tanto negativos como positivos. Sin embargo, como queremos calcular lo lejos que están del objetivo, es necesario aplicarles el valor absoluto para obtener comparaciones significativas. Como se muestra en la Tabla 2, N-BIR supera a los demás algoritmos siempre que la fracción de la prueba utilizada sea igual o superior al 40%.

4.2. INTERVALO DE PREDICCIÓN DE N-BIR

En esta sección, analizamos el IP generado por N-BIR. En la Fig. 5 mostramos tanto el MPIW como el PICP por porcentaje de tiempo utilizado. Como era de esperar, el MPIW disminuye a medida que aumenta el porcentaje utilizado para el entrenamiento. Sin embargo, en términos de PICP, se observa un comportamiento contrario a la intuición. Inicialmente, el PICP aumenta en la medida que lo hace el porcentaje, lo cual es de esperar. Sin embargo, a partir del 35-40%, empieza a disminuir. Esto puede explicarse por el hecho de que a medida que el MPIW se hace demasiado pequeño, es más probable que la temperatura final real esté fuera de los límites. Sin embargo, los errores absolutos medios observados son en su mayoría inferiores a 2 °C, por lo que esta disminución de la precisión no es significativa.

Si se analizan los resultados para un valor porcentual del 40% (como vimos en el subapartado anterior, este es el porcentaje en el que el algoritmo propuesto suministra mejores resultados), se observa un MPIW de 7,56°C y un PICP del 78,57%. En la Fig. 6 (izquierda), se muestra la distancia en °C entre el PI y la temperatura final de la prueba para los casos en los que la temperatura final estaba fuera del PI. Los valores negativos representan la distancia al límite inferior de las temperaturas reales finales por debajo del PI, mientras que los valores positivos representan la distancia al límite superior de las temperaturas finales por encima del PI. Como puede observarse, el error de la temperatura de la mayoría de los casos que quedan fuera del PI son inferiores a 5°C. En conclusión,

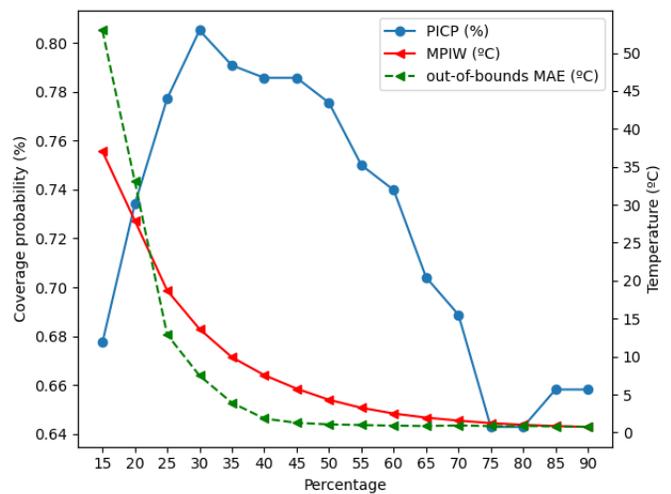


Fig. 5 PICP y MPIW por porcentaje generado por N-BIR.

el error absoluto medio fuera de los límites obtenido por el algoritmo utilizado para el 40% de la curva total es de 1,83°C (o -2,20°C para valores negativos, 0,93°C para valores positivos).

Si realizamos el mismo análisis, pero para instancias correspondientes al 50%, como se muestra en la Fig. 6 (derecha), obtenemos un MPIW de 4,30°C y un PICP del 77,55%. Las instancias fuera de los límites tienen un error absoluto medio de 1,07 °C (o -1,41°C para valores negativos, 0,57°C para valores positivos).

Como puede observarse, en ambos casos la distribución de errores no es normal, produciéndose más errores por debajo del límite inferior. Esto favorece la confianza en el resultado de que la temperatura no superará el límite máximo de la prueba BI.

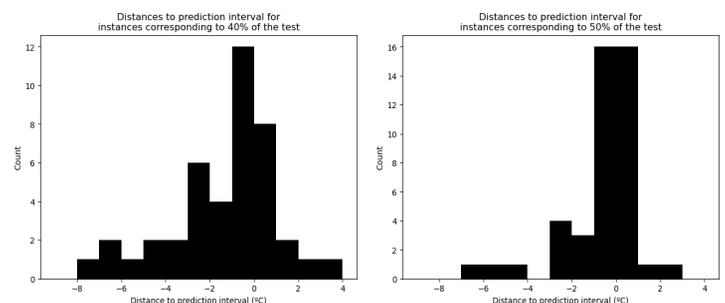


Fig. 6 Histograma del error de temperatura fuera de los límites.

5. DEBATE

Los resultados de este estudio demuestran que el algoritmo propuesto, N-BIR, es una herramienta prometedora para predecir la temperatura final de BI de los convertidores electrónicos de potencia, ya que puede reducir eficazmente el tiempo del test de BI actual en un 60%. En comparación con los predictores tradicionales basados en ML entrenados en un conjunto de datos, la naturaleza del algoritmo basada en la optimización numérica le permite basarse sólo en la instancia actual, sin necesidad de aprender de curvas pasadas, lo que resulta en una mayor flexibilidad. Además, la capacidad del algoritmo para proporcionar predicciones con un PI ayuda a determinar cuándo detener la prueba. Sin embargo, es importante señalar que la naturaleza del algoritmo, basada en la optimización numérica, también da lugar a un rendimiento deficiente para instancias pertenecientes a un porcentaje muy bajo de test utilizado (25% o menos). Reducir el test de BI a ese porcentaje no es realista.

También conviene tener en cuenta las limitaciones tanto de los modelos ML como de N-BIR. El rendimiento de los modelos ML se limita a las curvas pertenecientes a los modelos de convertidores utilizados para generar el conjunto de datos y puede no funcionar bien con datos pertenecientes a nuevos modelos con un comportamiento de calentamiento diferente. Por otro lado, la limitación de N-BIR no está relacionada con el conjunto de datos, sino con las hipótesis de partida formuladas, a partir de las cuales se obtuvo el modelo matemático del sistema. Hay algunos escenarios en los que estas hipótesis pueden no cumplirse, como en el caso de un sistema de refrigeración variable (dependiente de la temperatura actual del BI), un sistema de refrigeración más indirecto (con transmisores de calor intermedios entre el devanado a ensayar y el refrigerante), o un ensayo tan largo que se vea afectado por las condiciones climáticas.

La Tabla 3 muestra una comparación entre el algoritmo propuesto y otros trabajos presentados en la literatura. Como puede observarse, la capacidad de acortamiento del test del algoritmo propuesto es similar a la de otros métodos. Sin embargo, esta comparación debe tomarse con cautela, ya que cada uno de ellos utilizó datos de naturaleza diferente, lo que es un factor determinante en la complejidad del problema a abordar. Cabe destacar la capacidad del Análisis Paramétrico de pruebas propuesto en [4], que permite acortar el BI de 48h a 10h (79,17%). Sin embargo, este algoritmo tiene una limitación que N-BIR no tiene, sólo es capaz de acortar el test cuando aparecen fallos tempranos. Sin embargo, si no se detecta ningún fallo después de 10h, la prueba continúa hasta las 48h del test, y no se produce ninguna reducción de tiempo.

6. CONCLUSIONES Y TRABAJO FUTURO

En conclusión, el algoritmo propuesto, N-BIR, muestra un rendimiento prometedor para predecir las temperaturas finales de BI, proporcionando resultados precisos a la vez que reduce el tiempo de prueba. No sólo puede realizar predicciones precisas, sino que también puede generar un PI. El algoritmo puede reducir el tiempo de prueba de BI en un 40-50%, obteniendo al mismo tiempo predicciones altamente fiables. Además, el algoritmo se ha comparado con otros cinco algoritmos ML (LR, KNN, RF, XGBoost y CatBoost) y se ha demostrado que, siempre que se utilice al menos el 40% de la prueba total para predecir, supera a todos ellos en capacidad de predicción. Si se utiliza el 40% de la prueba para predecir la temperatura final, se obtiene un error medio esperado de $\pm 2,16^\circ\text{C}$ y un error máximo esperado de $\pm 12,56^\circ\text{C}$ (MPIW más el error máximo esperado). Si en cambio se utiliza el 50% de la curva, se obtiene un error medio esperado de $\pm 1,3^\circ\text{C}$ y un error máximo esperado de $\pm 6,3^\circ\text{C}$. Esto indica que utilizando el algorit-

Algorithm	BI time reduction	Observations
PCA [2]	50%	Accuracy: 90%
Flexible time reduction [15]	~55%	Sample size of 25k
Parametric Test Analysis [4]	79.17%	Only for failures
N-BIR	50%	MAE: 1.295°C
	60%	MAE: 2.164°C

Tabla 3 Comparación con otros métodos de la literatura.

mo podríamos acortar la duración de la prueba de BI un 50% o un 60%, siempre que los errores esperados antes mencionados sean aceptables. Este será el caso cuando la temperatura final prevista sumada al error máximo esperado sea inferior a la temperatura máxima necesaria para obtener un buen convertidor. No obstante, deben tenerse en cuenta las limitaciones del algoritmo y es necesario seguir investigando para validar su rendimiento en distintos escenarios y condiciones.

En cuanto a futuros trabajos, sería interesante comparar N-BIR con otros algoritmos de ML o incluso algoritmos de aprendizaje profundo (deep learning, DL) como las redes neuronales convolucionales o recurrentes (CNN, RNN). Además, el mismo estudio podría llevarse a cabo con un nuevo conjunto de datos para ver si estos resultados siguen siendo consistentes. Además, otro enfoque de investigación podría consistir en desarrollar un modelo matemático más complejo del sistema, que modele mejor el calentamiento de la prueba BI. No obstante, esto puede suponer un reto, ya que dependería del convertidor y requeriría un modelo diferente para cada convertidor. Por último, sería interesante aplicar este algoritmo en otros campos además de la electrónica de potencia para evaluar su eficacia.

REFERENCIAS

- Daasch, W. R.; Shirley, C. G.; Nahar, A. Statistics in Semiconductor Test: Going beyond Yield. *IEEE Design & Test of Computers*, vol. 26, no. 5, p. 64-73, Sept.-Oct. 2009, DOI: <https://doi.org/10.1109/MDT.2009.123>.
- Nahar, A.; Daasch, R.; Subramaniam, S. Burn-in reduction using principal component analysis. *IEEE International Conference on Test*, 2005., Austin, TX, USA, 2005, p. 10-155, DOI: <https://doi.org/10.1109/TEST.2005.1583971>.
- Stratigopoulos, H.; Drineas; Slamani, M.; Makris, Y. Non-RF to RF Test Correlation Using Learning Machines: A Case Study. *25th IEEE VLSI Test Symposium (VTS'07)*, Berkeley, CA, USA, 2007, pp. 9-14. ISSN: 1093-0167. DOI: <https://doi.org/10.1109/VTS.2007.41>.
- Sumikawa, N.; Wang, L. -C.; Abadir, M. S. An experiment of burn-in time reduction based on parametric test analysis. *2012 IEEE International Test Conference*, Anaheim, CA, USA, 2012, pp. 1-10. ISSN: 1089-3539. DOI: <https://doi.org/10.1109/TEST.2012.6401595>.
- Aydinkarahallıoglu C.; Jatar S.; Wang X.; Fong M.; Gupta V.; Troccoli M.; Hoffman AJ. Predicting early failure of quantum cascade lasers during accelerated burn-in testing using machine learning. *Sci Rep.* 2022 Jun 2;12(1):9184. ISSN 2045-2322 (online). DOI: <https://doi.org/10.1038/s41598-022-13303-0>
- HENG-TSAING TSENG & CHIEN-YU PENG. Optimal burn-in policy by using an integrated Wiener process, *IIE Transactions*, (2004) 36:12, 1161-1170, DOI: <https://doi.org/10.1080/07408170490507701>
- Lyu, Y. et al. Optimal Burn-in Strategy for High Reliable Products Using Convolutional Neural Network," in *IEEE Access*, vol. 7, pp. 178511-178521, 2019. ISSN: 2169-3536. DOI: <https://doi.org/10.1109/ACCESS.2019.2958570>.
- Kurz, D.; Lewitschnig, H.; Pilz, J. Flexible time reduction method for burn-in of high-quality products. *Quality and Reliability Engineering International*, vol. 37, no. 6, pp. 2900-2915, 2021.
- Steinier J, Termonia Y, Deltour J. Smoothing and differentiation of data by simplified least square procedure. *Anal Chem.* 1972 Sep 1;44(11):1906-9. DOI: <https://doi.org/10.1021/ac60319a045>. PMID: 22324618.
- Breiman, Leo. (1984). *Classification and Regression Trees* (1st ed.). Routledge. ISBN 9781315139470. <https://doi.org/10.1201/9781315139470>
- BREIMAN, Leo. Consistency for a simple model of random forests. *University of California at Berkeley. Technical Report*, 2004, vol. 670.
- Chen, T.; Guestrin, C. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '16)*. Association for Computing Machinery, New York, NY, USA, 785-794. <https://doi.org/10.1145/2939672.2939785>
- Friedman, J. H. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, vol. 29, no. 5, pp. 1189 - 1232, 2001.
- Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A. V.; GULIN, A. Catboost: unbiased boosting with categorical features. 2017. *Proceedings of the 32nd International Conference on Neural Information Processing Systems (NIPS'18)*. Curran Associates Inc., Red Hook, NY, USA, p. 6639-6649
- Coefficient of determination. (2023, April 4). In *Wikipedia*. https://en.wikipedia.org/wiki/Coefficient_of_determination
- Khosravi, Abbas & Nahavandi, Saeid & Creighton, Doug & Atiya, Amir. (2011). *Comprehensive Review of Neural Network-Based Prediction Intervals and New Advances*. *IEEE transactions on neural networks / a publication of the IEEE Neural Networks Council*. 22. 1341-56. 10.1109/TNN.2011.2162110.