

Mantenimiento predictivo de motores de buques mediante aprendizaje automático

Novoa Paradela, David^{1,*}; Eiras Franco, Carlos¹; Fontenla Romero, Óscar¹; Lamas López, Francisco²; Sanz Muñoz, David³

¹ CITIC, Universidad de A Coruña. Campus de Elviña s/n, 15071. A Coruña. david.novoa@udc.es, carlos.eiras.franco@udc.es, oscar.fontenla@udc.es

² CESADAR, Armada Española. Arsenal de Cartagena, 30290, Cartagena. flamlop@mde.es

³ DigitalLabs / IA, INDRA. Avenida de Bruselas 35, 28108. Madrid. dsmunoz@indra.es

* Autor principal

Resumen: La aparición de anomalías durante el funcionamiento de los sistemas industriales puede apuntar a la presencia de degradaciones y fallos, que deriven con el tiempo en comportamientos indeseados, pérdidas de operatividad y la rotura final del sistema. Las técnicas de mantenimiento predictivo se encargan de monitorizar el estado de los sistemas para llevar a cabo la detección de estas anomalías en fases incipientes, permitiendo programar las tareas de mantenimiento de forma óptima. En este trabajo se presenta una solución de mantenimiento predictivo para motores de buques basada en técnicas de inteligencia artificial. Para ello se hace uso de la información de los sensores (temperaturas, presiones, etc.) recogidos en tiempo real por los buques y transmitidos diariamente al centro de control. El sistema desarrollado es capaz de predecir la aparición de modos de fallos a partir de un histórico de datos del motor. Además, para que su uso sea escalable a grandes flotas, se ha implementado la solución en el entorno distribuido Spark.

Palabras clave: aprendizaje automático, mantenimiento predictivo, Spark.

1. Introducción

El coste de mantenimiento representa una parte importante de los costes operativos de la industria. En algunos casos, como en la industria metalúrgica, estos costes pueden llegar a suponer el 15%-60% de los costes totales de producción. Además, de estos, una tercera parte de la inversión se desperdicia como resultado de actividades innecesarias o incorrectas [1]. Sin embargo, el mantenimiento es crucial puesto que el fallo de un sistema puede llegar a suponer enormes costes económicos.

En el pasado, la imposibilidad de manejar los grandes y continuos flujos de datos han llevado a emplear, en muchos casos, técnicas estadísticas. El mantenimiento predictivo actual, sin embargo, sigue una filosofía más avanzada: En lugar de depender de estas estadísticas industriales (p. ej., tiempo medio entre fallos) para programar actividades de mantenimiento, se lleva a cabo una monitorización en tiempo real del sistema para determinar su estado. La capacidad de cómputo actual permite tanto procesar cantidades de datos mayores, así como la utilización de técnicas más sofisticadas para llevar a cabo predicciones, detección de estados anormales y un diagnóstico del sistema. Por tanto, el mantenimiento predictivo se puede entender como un mantenimiento preventivo [2] condicionado al estado actual del sistema y a las predicciones a futuro realizadas a partir de un histórico de operación.

En este trabajo de investigación se presenta el desarrollo de un sistema de mantenimiento predictivo enmarcado dentro del proyecto SOPRENE en su aplicación a los motores de buques de la armada. El sistema planteado ha analizado y empleado las técnicas de aprendizaje automático en entornos distribuidos. En este sentido, las metodologías consideradas se pueden dividir según lo expuesto por Ran *et al.* [3]:

1.1 Categorías en función del propósito

En base al criterio de optimización seguido, podemos distinguir entre:

- Minimización de costes: la métrica de coste empleada es habitualmente el tiempo de vida útil (RUL, *Remaining Useful Life*) del sistema, aunque también es posible definir un modelo de coste *ad hoc* [4].
- Maximización de la fiabilidad y la disponibilidad: las métricas son la probabilidad de un sistema de encontrarse en un estado de funcionamiento normal dado un intervalo de tiempo [5] y la probabilidad de que el sistema se encuentre operativo [6].
- Optimización multiobjetivo: se busca optimizar múltiples métricas simultáneamente para lograr un mejor equilibrio entre los objetivos. Además de las antes mencionadas, emplean métricas como el riesgo, la seguridad o la viabilidad. Generalmente, es imposible obtener los valores óptimos para todos los objetivos al mismo tiempo, por lo que se han desarrollado una gran variedad de modelos multiobjetivo [7]-[9].

1.2 Categorías en función de la aproximación

En base al tipo de aproximación empleada, podemos distinguir entre:

- Aproximaciones basadas en conocimiento. Se utiliza conocimiento experto y procesos de razonamiento deductivo. Existen aproximaciones basadas en ontologías [10], en reglas [11] o en modelos que tratan de vincular los procesos físicos de un sistema con modelos matemáticos, como modelos Gaussianos [12], modelos de sistemas lineales [13] o modelos de Markov [14].
- Aproximaciones basadas en técnicas clásicas de aprendizaje automático (*machine learning*, ML). Se han utilizado redes de neuronas artificiales [15], los árboles de decisión [16] (incluyendo el algoritmo Random Forest [17]). así como máquinas de soporte de vectores (SVM), tanto de forma supervisada [18] como no supervisada [19]. Por último, la técnica de vecinos más cercanos (k-NN) es una de las más comunes para la clasificación de fallos [20], para la predicción de tiempo de vida útil (RUL) [21] y la detección temprana [22].

- Aproximaciones basadas en aprendizaje profundo (*deep learning*). Una de las más empleadas son las redes neuronales Autoencoder, cuya capa de salida busca reproducir los datos presentados en su capa de entrada después de haber pasado por una fase de compresión dimensional, permitiendo crear modelos robustos ante el ruido [23]. También se han utilizado redes neuronales recurrentes (RNN), basadas en celdas Long Short Term Memory Networks (LSTM) [24], que pueden aprender dependencias a más largo plazo. Este tipo de redes son muy potentes para el análisis de secuencias [25].

2. Contextualización del problema

La aplicación de las técnicas de mantenimiento predictivo a los motores de los buques de la Armada está definida por las características de los mismos: Los buques considerados registran cada 10 s los valores de cerca de 300 variables asociadas al motor, representando una amplia variedad de aspectos físicos del mismo (e.g., temperatura del gas, presión en los filtros, etc). y se dispone de 4 años de históricos de operación de 4 buques. Sin embargo, los datos registrados presentan una calidad heterogénea debido a fallos en los sensores (valores erróneos o perdidos) o problemas en las comunicaciones (frecuencias de muestreo no uniformes). Por otro lado, existen tres modos de funcionamiento del motor en función de las revoluciones por minuto (RPM) a las que trabaja: motor apagado (RPM cercanas a cero), motor a ralentí (RPM aproximadas a un umbral μ) y motor en operación (RPM superiores a un umbral μ), teniendo que caracterizar y filtrar los estados de interés.

En cuanto a los fallos de funcionamiento en el histórico de datos, debido a la extensión de los archivos, su identificación y etiquetado manual por parte de un experto es inviable. De este modo, no se dispone de un conjunto de modos de fallo localizados y, por tanto, la detección de anomalías debe ser desarrollada de forma no supervisada. Para suplir esta carencia se utiliza un análisis FMECA (*Failure Mode, Effects, and Criticality Analysis*) que describe de forma teórica los modos de fallo que se pueden producir durante el funcionamiento del motor (las variables o elementos que intervienen en dichos modos y sus valores nominales, máximos y mínimos).

Finalmente, debido a que la solución propuesta no debe analizar un único buque, sino una flota, la gestión y control se lleva a cabo de manera centralizada. Cada buque envía los datos registrados por sus sensores a un nodo central donde se almacenan y procesan. Ya que los conjuntos de datos con los que se va a tratar pueden llegar ser de gran tamaño, y para que el sistema sea escalable a un número elevado de buques, el sistema ha de ser distribuido: se ha utilizado el sistema de ficheros distribuido HDFS para almacenar los datos y el entorno Apache Spark para entrenar y ejecutar los modelos de forma distribuida.

3. Solución propuesta

La solución diseñada combina las soluciones a varios subproblemas. Así, la arquitectura general se compone de cuatro bloques o tareas claramente delimitadas (véase Figura 1):

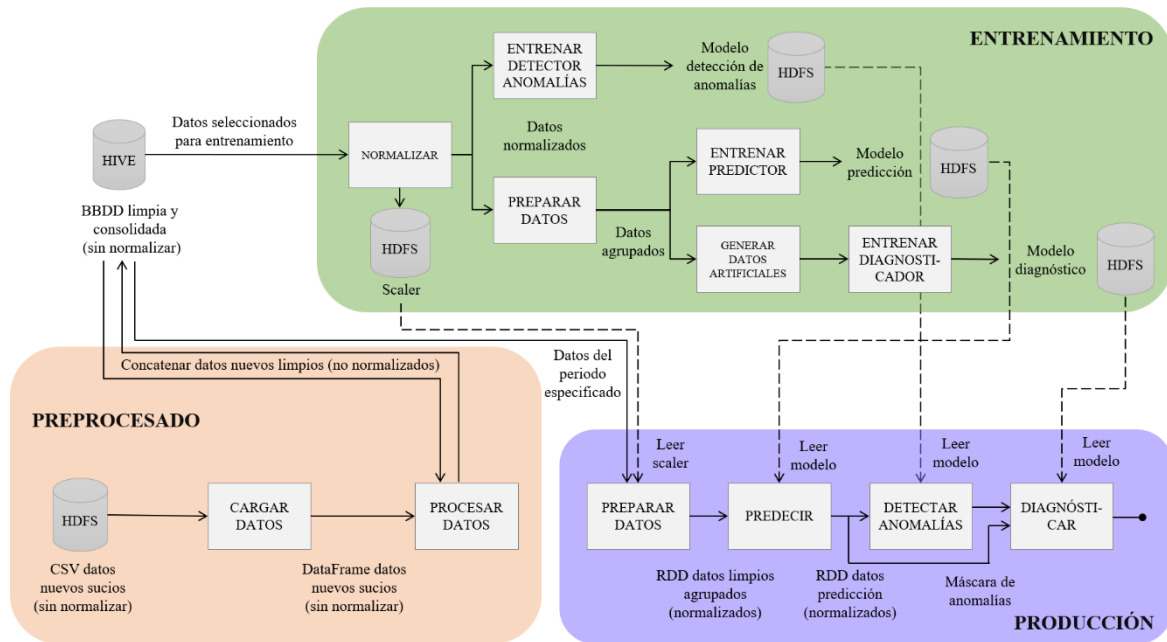


Figura 1. Arquitectura de la solución.

3.1 Módulo de preprocesamiento

Para entrenar los modelos de ML se necesita disponer de un conjunto de datos robusto y representativo de los diferentes estados del motor. Los valores medidos por los sensores del sistema se encuentran almacenados en ficheros CSV en HDFS, pero, como se ha detallado en la sección 3, estos datos precisan ser preprocesados. Así, en este bloque se comprueba la existencia de valores perdidos, se unifica la frecuencia de muestreo y lleva a cabo un proceso de selección de variables:

- *Cargar datos:* El flujo comienza mediante la lectura del conjunto de datos que se desea procesar y que se encuentran almacenados en HDFS.
- *Procesar datos:* Para solventar el problema de la baja calidad de los datos originales (sección 3) se llevan a cabo dos acciones:
 1. Selección de variables: de todas las variables se emplearán aquellas que presentan suficiente variabilidad en sus valores. Esta selección se lleva a cabo tanto automáticamente (desechando variables que presentan valores constantes) y manualmente, eliminando las seleccionadas por el usuario.
 2. Estandarización de las frecuencias: para evitar el segundo problema es necesario homogeneizar la frecuencia de muestreo. Así, se crea, a partir de los datos iniciales, un conjunto en el que exista una medida simultánea para todas las variables cada 60 s. Para ello se rellena la ausencia de valores de las variables que presenten una frecuencia inferior con el último valor disponible.

Los datos procesados son almacenados en una base de datos consolidada de Hive.

- *Normalizar datos:* Debido a que los valores recogidos por los sensores oscilan en rangos muy diferentes entre sí para cada variable se ejecuta un proceso de normalización de forma individual. El normalizador entrenado se almacena en HDFS para ser utilizado en producción con la llegada de nuevos datos.

- *Preparar datos:* Para dotar de flexibilidad al sistema, el usuario puede establecer la unidad del horizonte con el que llevar a cabo la predicción (horas, días, semanas o meses), de forma que los datos normalizados se agrupen temporalmente (ej. si se van a realizar predicciones en días, los datos normalizados se agrupan en un único dato por día). Además, antes de agrupar los datos, estos se filtran en función del valor de la variable RPM para eliminar aquellos que se correspondan con instantes del motor apagado.

3.2 Módulo de predicción

El objetivo principal es conocer el estado del motor en un instante de tiempo futuro. De esta manera, para llevar a cabo una predicción desde un instante de tiempo t_i , para un instante futuro que dista a *horizonte* unidades de tiempo de uso del motor ($t_{i+horizonte}$), el sistema deberá recibir la información recogida por los sensores durante las últimas *ventana* unidades de tiempo anteriores a t_i ($t_{i-ventana}$), definida junto al horizonte por el usuario. En la Figura 2 se muestra una representación gráfica de estos conceptos.

Los métodos de predicción disponibles son: regresión lineal (en su versión distribuida), y de redes LSTM y Elastic Net (de forma no distribuida). Tras el entrenamiento de los modelos, estos son almacenados en HDFS para ser usados en producción del mismo modo que los normalizadores.

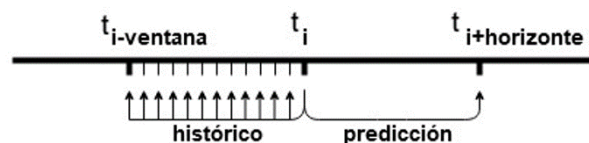


Figura 2. Utilización de una ventana de datos previos para llevar a cabo una predicción.

3.3 Módulo de detección de anomalías

Una vez realizada la predicción del estado del motor, es necesario determinar si dicho estado se corresponde con un valor normal o no. Al no disponer de anomalías etiquetadas, el proceso de detección se realiza de forma no supervisada utilizando una red Autoencoder. Mediante el error de reconstrucción podemos discernir entre datos normales (bajo error de reconstrucción) y datos anómalos (errores altos). Este valor puede presentarse como el error cuadrático medio de todas las variables de entrada (un único valor) o su descomposición, el error de cada una de las variables o nodos de entrada de la red. El objetivo, dado un conjunto de registros, es determinar cuáles son anómalos y qué variables provocan dichas anomalías, lo que se hace en tres subfases secuenciales:

1. Detectar anomalías: para determinar qué registros son anómalos se realiza un primer filtrado mediante el error cuadrático medio. A partir de un error umbral precalculado se clasifican los datos que lo superen como anómalos y el resto como normales. El usuario escoge si el cálculo de este error umbral se realiza mediante el rango intercuartílico o estableciendo un porcentaje de datos anómalos en el conjunto.
2. Independizar contribuciones: para determinar qué variables han sido las causantes de la aparición de la anomalía en los datos clasificados como anómalos, se emplea la

descomposición del error de reconstrucción. Así, se pasa de un único error global a tantos como variables conformen el registro, pudiendo ordenar las variables por su error de reconstrucción y determinar de forma automática la contribución de cada variable utilizando el método Elbow [26].

3. Construir una máscara de anomalías: a partir de la selección de la subfase anterior, se construye una matriz o máscara de salida de dimensiones $m \times n$ (siendo m el número de filas o registros y n el número de columnas o variables) en la que las variables anómalas se marcan con un uno y las normales con un cero.

3.4 Módulo de diagnóstico

El bloque de diagnóstico es el encargado de, a partir de la predicción y la máscara de detección de anomalías, determinar qué modos de fallo pueden producirse y su probabilidad. Para determinar la probabilidad de cada modo de fallo se ha decidido utilizar un modelo de clasificación supervisado basado en redes neuronales. Debido a que no se dispone de conjuntos de datos etiquetados de todos los posibles modos de fallo, se ha implementado un generador de datos artificiales que los produzca a partir de las características teóricas de los modos de fallo recogidas en el documento FMECA (variables implicadas, valores nominales, valores tope de rango, etc.). Así, el modelo es entrenado considerando a cada uno de los modos de fallo como una clase de salida, teniendo en su capa de entrada tantas neuronas como variables del motor, y en su capa de salida tantas *softmax* como modos de fallo.

El clasificador de modos de fallo se emplea en combinación con la máscara de salida del bloque de detección de anomalías para llevar a cabo el diagnóstico del motor, utilizada para acotar los modos de fallo posibles, omitiendo los modos de fallo que todas sus variables implicadas han sido consideradas normales. La salida de este módulo (y final del sistema) contiene el identificador del modo de fallo en el FMECA, la fecha prevista en la que se produzca el modo de fallo y la probabilidad/certidumbre asociada a la misma.

4. Resultados

Dado que no se dispone de un conjunto de *targets* o valores de referencia, no ha sido posible realizar una evaluación cuantitativa del sistema. El desarrollo de las distintas soluciones parciales que conforman la solución final ha sido supeditado a una evaluación mayormente cualitativa por los expertos de la organización implicada:

- Predicción: el filtrado de los datos por RPM y su agrupación reduce considerablemente el número de ejemplos disponibles. Esto incapacita a modelos como las redes LSTM que precisan grandes cantidades de datos. En estos escenarios se ha visto que los modelos más sencillos reportan mejores resultados, siendo posible realizar predicciones hasta 10 días/semanas/meses con una calidad razonable.
- Detección de anomalías: Los resultados del detector de anomalías se han comparado con las alarmas registradas en los buques a lo largo de los cuatro años. Tal y como se observa en la Figura 3, el modelo fue capaz de detectar la mayoría de estas anomalías e incluso anticipar la ocurrencia de algunas de ellas.
- Diagnóstico: la generación de conjuntos de datos artificiales basados en el FMECA ha permitido construir modelos de clasificación que determinen qué modos de fallo

se están produciendo. Sin embargo, este comportamiento teórico del motor no tiene por qué corresponderse siempre con la realidad, ya que su funcionamiento puede variar con el uso, reemplazamiento o la reparación de piezas.

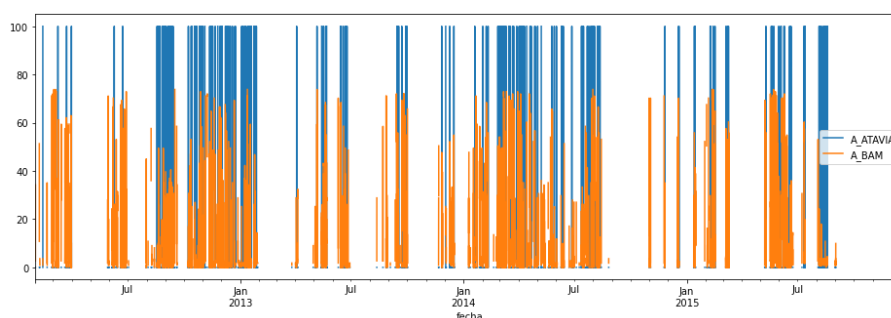


Figura 3. Estados anómalos etiquetados manualmente por un experto humano (azul) frente a los detectados por el sistema (naranja).

5. Conclusión

La solución desarrollada permite predecir la aparición de los diferentes modos de fallo descritos en el FMECA del motor de combustión de un buque. Las tareas de predicción y detección de anomalías son totalmente independientes, por lo que esta última se puede llevar a cabo tanto para momentos futuros (datos provenientes de la predicción), presentes (tiempo real) o pasados (análisis a posteriori). La mayor responsabilidad recae en la tarea de predicción ya que las operaciones posteriores parten de la salida de este módulo. Para dotarlo de flexibilidad se proporciona un amplio abanico de métodos, fácilmente extensible si es preciso, a la vez que se permite al usuario configurar los parámetros de predicción para obtener los resultados más adecuados en cada situación. El sistema es altamente configurable y su uso se puede extrapolar a otros buques que presenten características similares. Su ejecución es distribuida, por lo que los tiempos de predicción, detección y diagnóstico son bajos, siendo el mayor cuello de botella la tarea de preprocesado, operación que no se lleva a cabo de manera distribuida.

Agradecimientos

Queremos agradecer a la Armada Española la colaboración en el proyecto SOPRENE, así como a la Consejería de Educación, Universidad y Formación Profesional de la Xunta de Galicia, por la financiación ofrecida al CITIC a través del Fondo Europeo de Desarrollo Regional (FEDER) y de la Secretaría General de Universidades (Ref. ED431G 2019/01).

Referencias

- [1] R. K. Mobley, *An Introduction to Predictive Maintenance*, 2nd ed., Elsevier, 1990.
- [2] O. Motaghare, A. S. Pillai, and K. I. Ramachandran, "Predictive maintenance architecture," *IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, 2018, pp. 1–4.
- [3] Y. Ran, X. Zhou, P. Lin, Y. Wen, and R. Deng, "A survey of predictive maintenance: Systems, purposes and approaches," *arXiv:1912.07383*, 2019.
- [4] Y. He, X. Han, C. Gu, and Z. Chen, "Cost-oriented predictive maintenance based on mission reliability state for cyber manufacturing systems," *Advances in Mechanical Engineering*, 2018.

- [5] S. Song, D. W. Coit, and Q. Feng, "Reliability analysis of multiple-component series systems subject to hard and soft failures with dependent shock effects," *IIE Transactions*, vol. 48, no. 8, pp.720–735, 2016.
- [6] M. A. Gravette and K. Barker, "Achieved availability importance measure for enhancing reliability-centered maintenance decisions," *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, vol. 229, no. 1, pp. 62–72, 2015.
- [7] L. Lin, B. Luo, and S. Zhong, "Multi-objective decision-making model based on CBM for an aircraft fleet with reliability constraint," *International Journal of Production Research*, vol. 56, no. 14, pp. 4831–4848, 2018.
- [8] J. Zhao and L. Yang, "A bi-objective model for vessel emergency maintenance under a condition-based maintenance strategy," *Simulation*, vol. 94, no. 7, pp. 609–624, 2018.
- [9] Y. Xiang, D. Coit, and Z. Zhu, "A multi-objective joint burn-in and imperfect condition-based maintenance model for degradation-based heterogeneous populations," *Quality and Reliability Engineering International*, vol. 32, no. 8, pp. 2739–2750, 2016.
- [10] A. Konys, "An ontology-based knowledge modelling for a sustainability assessment domain," *Sustainability*, vol. 10, 300, 2018.
- [11] M. D. Ying Peng and M. J. Zuo, "Current status of machine prognostics in condition-based maintenance: a review," *The International Journal of Advanced Manufacturing Technology*, vol. 50, pp. 297–313, 2010.
- [12] N. Chen, Z.-S. Ye, Y. Xiang, and L. Zhang, "Condition-based maintenance using the inverse Gaussian degradation model assessment domain," *European Journal of Operational Research*, vol. 243, no. 1, p. 190–199, 2015.
- [13] A. Lucifredi, C. Mazzieri, and M. Rossi, "Application of multiregressive linear models, dynamic kriging models and neural network models to predictive maintenance of hydroelectric powers systems," *Mechanical Systems and Signal Processing*, vol. 14, no. 3, pp. 471–494, 2000.
- [14] M. Calder and M. Sevegnani, "Stochastic model checking for predicting component failures and service availability," *IEEE Transactions on Dependable and Secure Computing*, vol. 16, no. 1, pp. 174–187, 2019.
- [15] B. Samanta and K. Al-Balushi, "Artificial neural network based fault diagnostics of rolling element bearings using time-domain features," *Mechanical Systems and Signal Processing*, vol. 17, pp. 317–328, 2003.
- [16] J. R. Quinlan, "Improved use of continuous attributes in C4.5," *CoRR*, vol. cs.AI/9603103, 1996. [Online]. Available: <https://arxiv.org/abs/cs/9603103>
- [17] J. Shi, N. Niu, and X. Zhu, "A fault diagnosis method for multi-condition system based on random forest," *Prognostics and System Health Management Conference (PHM-Paris)*, pp. 350–355, 2019.
- [18] Y. Chen, H. Yigang, Z. Li, L. Chen, and C. Zhang, "Remaining useful life prediction and state of health diagnosis of lithium-ion battery based on second-order central difference particle filter," *IEEE Access*, vol. 8, pp. 37305–37313, 2020.
- [19] G. Ratsch, S. Mika, B. Scholkopf, and K. Muller, "Constructing boosting algorithms from SVMs: an application to one-class classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 9, pp. 1184–1199, 2002.
- [20] G. A. Susto, A. Schirru, S. Pampuri, S. McLoone, and A. Beghi, "Machine learning for predictive maintenance: A multiple classifier approach," *IEEE Transactions on Industrial Informatics*, vol. 11, no. 3, pp. 812–820, 2015.
- [21] Z. Liu, W. Mei, X. Zeng, C. Yang, and X. Zhou, "Remaining useful life estimation of insulated gate bipolar transistors (IGBTs) based on a novel volterra k-nearest neighbor optimally pruned extreme learning machine (VKOPP) model using degradation data," *Sensors*, vol. 17, no. 11: 2524, 2017.
- [22] X. long Chen, P. hong Wang, Y. sheng Hao, and M. Zhao, "Evidential KNN-based condition monitoring and early warning method with applications in power plant," *Neurocomputing*, vol. 315, pp. 18 – 32, 2018.
- [23] M. Ma, C. Sun, and X. Chen, "Deep coupling autoencoder for fault diagnosis with multimodal sensory data," *IEEE Transactions on Industrial Informatics*, vol. 14, pp. 1137–1145, 2018.
- [24] J. Yuan and Y. Tian, "An intelligent fault diagnosis method using GRU neural network towards sequential data in dynamic processes," *Processes*, vol. 7, no. 3: 152, 2019.
- [25] R. Yang, M. Huang, Q. Lu, and M. Zhong, "Rotating machinery fault diagnosis using long-short-term memory recurrent neural network," *IFAC-Papers OnLine*, vol. 51, no. 24, pp. 228–232, 2018, 10th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes (SAFEPROCES).
- [26] P. Bholowalia and A. Kumar, "Article: Ebc-means: A clustering technique based on elbow method and k-means in WSN," *International Journal of Computer Applications*, vol. 105, no. 9, pp. 17–24, 2014.