

Detección y Aislamiento de Fallas en una Red de Termopares Mediante Técnicas Basadas en Datos

Diego Alexander Velandia Cárdenas

Universidad Nacional de Colombia
Facultad de Ingeniería
Departamento de Ingeniería Eléctrica y Electrónica
Bogotá, Colombia
2022

Detección y Aislamiento de Fallas en una Red de Termopares Mediante Técnicas Basadas en Datos

Diego Alexander Velandia Cárdenas

Trabajo de grado presentado como requisito parcial para optar al título de:
Maestría en Ingeniería - Automatización Industrial

Director:
Ph.D.Jorge Iván Sofrony Esmeral

Codirector:
Ph.D Erwin José López Pulgarín

Universidad Nacional de Colombia
Facultad de Ingeniería
Departamento de Ingeniería Eléctrica y Electrónica
Bogotá, Colombia
2022

Lista de Figuras

1-1 Diagrama de flujo (parcial) del proceso de obtención de ferroníquel 6
1-2 Ubicación relativa de *Plate Coolers* y *Waffle Coolers* 9
1-3 Distribución de secciones angulares del horno [1] 10
1-4 Ubicación de paneles en la pared del horno. 10
1-5 Esquema básico de un termopar sometido a una diferencia de temperatura. . 12
1-6 Ejemplo de sensor con señal anclada. 13
1-7 Ejemplo de evolución de sensor con fallo intermitente. 14
1-8 Diagrama de flujo de aproximaciones basadas en modelo 17
1-9 Redundancia física y redundancia analítica 18
1-10 Diagrama de flujo de aproximaciones basadas en conocimiento 21
1-11 Ejemplo de clustering usando *k-means* 23
1-12 Diagrama de flujo de un modelo de *Machine Learning* 23
1-13 Diagrama de flujo para la obtención de un modelo de *Machine Learning* de aprendizaje supervisado 25
1-14 Esquema de árbol de decisión 28
1-15 Esquema de división de datos en entrenamiento *k-folds* 30
1-16 Esquema de ejemplo de caso de *underfitting* 32
1-17 Esquema de ejemplo de caso de *overfitting* 32
1-18 Esquema de ejemplo de caso de *fitting* 33
1-19 Estructura de matriz de confusión para *label* de 2 categorías 34

4-1 Diagrama de flujo de proceso de limpieza de datos 38

5-1 Ejemplo de matriz de correlación (izq) y representación en mapa de calor (der). 43
5-2 Ubicación de paneles en posición 12 44
5-3 Extracto de línea de tiempo de paneles en la posición angular 12 45
5-4 Matriz de correlación de paneles en la posición angular 12 45
5-5 Plots de radar de *Plate Coolers* 46
5-6 Plots de radar de *Waffle Coolers* 47
5-7 Matriz de correlación de *Waffle Coolers* 48
5-8 Acercamiento de matriz de correlación de *Waffle Coolers* 49
5-9 *Waffle Coolers* agrupados entre posiciones angulares 30 y 50 50
5-10 Análisis descriptivo de variables de vecindario alrededor de *Waffle Cooler* 33 A 50
5-11 Línea de tiempo de variables alrededor de *Waffle Cooler* 33 A 51

5-12 Comparación de matrices de correlación de vecindario en torno al *Waffle Cooler 33 A* 51
5-13 Comparación de matrices de correlación de vecindario en torno al *Waffle Cooler 33 A* con líneas de tiempo acotadas a 2017 y 2018 52
5-14 Matriz de correlación *Plate Coolers* 53
5-15 Matriz de correlación de *Plate Coolers* por nivel 54
5-16 Comparación de matrices de correlación de *Plate Coolers* alrededor del panel 43 A con filtrado de datos 55
5-17 Ejemplo de vecindario de termopares adyacentes 55
5-18 Posición angular de vecindario de termopares adyacentes 56
5-19 Matriz de correlación de vecindario de termopares adyacentes 56
5-20 Matriz de correlación de vecindario de termopares adyacentes con datos filtrados 57
5-21 Ejemplo de fallo de corta duración 58
5-22 Ejemplo de división de un conjunto de datos en ventanas 59
5-23 Ubicación de termopares de vecindario de *Waffle Coolers*. 61
5-24 Análisis de correlación en ventanas de tiempo de 1 hora 61
5-25 Análisis de correlación en ventanas de tiempo de 6 horas 61
5-26 Análisis de correlación en ventanas de tiempo de 12 horas 62
5-27 Análisis de correlación en ventanas de tiempo de 1 día 62
5-28 Análisis de correlación en ventanas de tiempo de 3 días 62
5-29 Análisis de correlación en vecindario de *Waffle Coolers*. 63
5-30 Diferencia de temperatura en vecindario de *Waffle Coolers*. 63
5-31 Temperatura de *Waffle Coolers* 69 A y 70 A alrededor del 01-07-2019. 64
5-32 Detección de datos anclados por varianza 66
5-33 Lecturas del *Waffle Cooler* 56 B alrededor de fechas detectadas 67
5-34 Detección de outliers en ventanas de tiempo 68
5-35 Comportamiento de outliers del *Waffle Cooler* 29 B alrededor del intervalo de interés 70
5-36 Variación de detección de ventanas con outliers según ancho de ventana . . . 72
5-37 Función de distribución de probabilidad y *CDF* para datos con distribución normal estándar 73
5-38 Ejemplo de comparación de dos *CDF*s y medida de la distancia entre las distribuciones 73
5-39 Diagrama de flujo de cálculo de *CDF* por tipo, nivel y cuadrante de panel . 74
5-40 Comparación de *CDF*s de *Plate Coolers* de nivel A para los 4 cuadrantes del horno 75
5-41 Comparación de *CDF*s de *Plate Coolers* de nivel B para los 4 cuadrantes del horno 75
5-42 Comparación de *CDF*s de *Plate Coolers* de nivel C para los 4 cuadrantes del horno 76

5-43 Comparación de *CDF*s de *Plate Coolers* de nivel D para los 4 cuadrantes del horno . . . 76
5-44 Comparación de *CDF*s de *Waffle Coolers* para los 4 cuadrantes del horno . . 77
5-45 *Plate Coolers* 34 D a 36 D antes y después de mantenimiento . . . 78
5-46 *CDF* del *Plate Cooler* 35 D antes del mantenimiento . . . 79
5-47 *CDF* del *Plate Cooler* 35 D después del mantenimiento . . . 79
5-48 *CDF* del *Plate Cooler* 34 D antes del mantenimiento . . . 80
5-49 *CDF* del *Plate Cooler* 36 D antes del mantenimiento . . . 80
5-50 *CDF* del *Plate Cooler* 34 D después del mantenimiento . . . 81
5-51 *CDF* del *Plate Cooler* 36 D después del mantenimiento . . . 81
5-52 Ejemplo de segmento con fallo identificado . . . 84

6-1 Estructuras de modelos de Árbol de decisión (izq.) y Bosque aleatorio (der.) 87
6-2 Ejemplo de estructura de una red neuronal . . . 88
6-3 Esquema de una neurona . . . 89
6-4 Esquema de división de dos grupos de observaciones mediante dos hiperplanos H_1 y H_2 . . . 91
6-5 Diseño experimental fase 1 . . . 95
6-6 Diseño experimental fase 2 . . . 95

7-1 Segmentos de datos adyacentes de *Plate Coolers* del nivel C . . . 96
7-2 Diagrama de flujo de cálculo de *features* . . . 97
7-3 Esquema de movimiento de ventana deslizante . . . 97
7-4 Ejemplo de matriz de confusión . . . 100
7-5 Esquema de matriz de confusión . . . 101
7-6 Ejemplo de matriz de confusión normalizada . . . 101
7-7 Ejemplo de curvas de aprendizaje con buen ajuste . . . 102
7-8 Ejemplo de curvas de aprendizaje con ajuste invariante . . . 103
7-9 Ejemplo de curvas de aprendizaje con sobreajuste . . . 104
7-10 Curvas de aprendizaje *hold-out* modelo *Random Forest* con *Plate Coolers B* 106
7-11 Curvas de aprendizaje *hold-out* modelo *Random Forest* con *Plate Coolers D* 107
7-12 Curvas de aprendizaje *hold-out* modelo *XGBoost* con *Plate Coolers B* . . . 107
7-13 Curvas de aprendizaje *hold-out* modelo *XGBoost* con *Plate Coolers C* . . . 108
7-14 Curvas de aprendizaje modelo *XGBoost* ventana de 24 horas, 5 paneles, sin *early stopping* . . . 109
7-15 Curvas de aprendizaje modelo *XGBoost* ventana de 24 horas, 5 paneles, con *early stopping* . . . 109
7-16 Curvas de aprendizaje modelo *XGBoost* ventana de 12 horas, 5 paneles, sin *early stopping* . . . 111
7-17 Curvas de aprendizaje modelo *XGBoost* ventana de 12 horas, 3 paneles, con *early stopping* . . . 111

7-18 Vecindario de *Waffle Coolers* de validación 112
7-19 Diagrama de flujo de validación ciega 113
7-20 Extracto de *labels* etiquetados con modelo de ancho de ventana de 6 horas . 114
7-21 Diagrama de flujo de algoritmo de contador *up-down* reducido 115
7-22 Cambio en sensibilidad para modelos de *Machine Learning* con filtros *up-down* 116
7-23 Cambio en especificidad para modelos de *Machine Learning* con filtros *up-down* 117
7-24 Ventana en fallo 1 detectada en validación ciega 119
7-25 Ventana en fallo 2 detectada en validación ciega 120

Lista de Tablas

1-1 Resultados de matriz de confusión de dos categorías 33

4-1 Proceso de eliminación de TAGs en dataset original con base en los criterios de calidad definidos. 39
4-2 Conteo y dristribución de TAGs de temperatura de *Waffle Coolers* y *Plate Coolers* 41

5-1 Anchos de ventana y cantidad equivalente de registros individuales 60
5-2 Conteo de ventanas congeladas de ancho 1 día para termopares del horno . . 66
5-3 Comparación de detección de valores anclados con ventanas de tiempo de diferente ancho 67
5-4 Conteo de ventanas con outliers, ventana 1 día y umbral 5 para termopares del horno 69
5-5 Caracterización de ventanas de tiempo en dataset 70
5-6 Comparación de detección de ventanas con outliers en ventanas de tiempo de diferente ancho 71
5-7 Conteo de registros y fallos detectados con fallos mediante inspección visual . 84

6-1 Modelos de Machine Learning de aprendizaje supervisado 86
6-2 Parámetros y valores 94

7-1 Tamaños de muestra por tipo de panel y ancho de ventana 99
7-2 Resultados de entrenamiento de modelos para arquitectura de **modelo único** 105
7-3 Resultados preliminares entrenamiento por tipo y nivel en arquitectura **por tipo y nivel** 106
7-4 Resultados diseno experimental fase 2 110
7-5 Resultados validación ciega 113
7-6 Comparación resultados de modelos con filtro *up-down* 118
7-7 Varianza ventana en fallo 1 detectada en validación ciega 119
7-8 Varianza ventana en fallo 2 detectada en validaci\'on ciega 120

Contenido

Lista de Figuras III

Lista de tablas VII

1 Introducción 5
- 1.1 Estructura del documento . . . 5
- 1.2 La planta y el proceso . . . 6
  - 1.2.1 Enfriamiento de las paredes laterales del horno . . . 8
  - 1.2.2 Detección e identificación de fallos en termopares del horno . . . 11
- 1.3 Termopares . . . 11
  - 1.3.1 Clasificación de fallos en termopares . . . 13
- 1.4 Detección y Aislamiento de Fallos (*FDI*) . . . 15
- 1.5 Aproximaciones Basadas en Modelos . . . 17
- 1.6 Aproximaciones Basadas en Conocimiento . . . 20
- 1.7 Aproximaciones Basadas en Datos (*Data-Driven*) . . . 22
  - 1.7.1 Aprendizaje no supervisado . . . 23
  - 1.7.2 Aprendizaje supervisado . . . 23
    - 1.7.2.1 Establecimiento de *labels* y *features* . . . 25
    - 1.7.2.2 Selección del modelo . . . 27
    - 1.7.2.3 Entrenamiento del modelo . . . 29
    - 1.7.2.4 Validación del modelo . . . 33

2 Identificación del problema 35
- 2.1 Identificación del problema . . . 35

3 Objetivos 36
- 3.1 Objetivo General . . . 36
- 3.2 Objetivos Específicos . . . 36

4 Caracterización y preprocesamiento del conjunto de datos 37
- 4.1 Metodología . . . 37
- 4.2 Resultados y discusión . . . 39

5 Generación de métricas 42
5.1 Uso de termopares cercanos como alternativa a redundancia física 43
5.1.1 Metodología 43
5.1.2 Vecindarios de paneles agrupados por posición angular 44
5.1.3 Vecindarios de paneles agrupados por tipo y nivel de panel 46
5.1.3.1 Matrices de correlación de *Waffle Coolers* 47
5.1.3.2 Matriz de Correlación de *Plate Coolers* 52
5.1.4 Vecindarios de paneles agrupados por adyacencia a un panel central . 55
5.1.5 Resultados y discusión 57
5.2 Análisis en ventanas de tiempo 58
5.2.1 Metodología 58
5.2.2 Resultados y discusión 60
5.3 Detección heurística de valores anclados en ventanas de tiempo 65
5.3.1 Metodología 65
5.3.2 Resultados y discusión 66
5.4 Detección heurística de outliers 68
5.4.1 Metodología 68
5.4.2 Resultados y discusión 69
5.5 Función de Distribución Acumulativa (*CDF*) y presencia de fallos 72
5.5.1 Metodología 74
5.5.2 Resultados y discusión 75
5.5.3 Distancia entre *CDF*s para termopares del horno 77
5.6 Selección de métricas 82
5.7 Establecimiento de línea base 83
5.7.1 Metodología 83
5.7.2 Resultados y discusión 83

6 Diseño de metodología de FDI 85
6.1 Selección de modelos de *Machine Learning* 85
6.1.1 Bosque Aleatorio (*Random Forest*) 86
6.1.2 Redes Neuronales (*Neural Networks*) 88
6.1.3 Support Vector Machines (SVM) 90
6.1.4 *Extreme Gradient Boosting (XGboost)* 91
6.2 Selección de arquitectura y modo de entrenamiento 93
6.3 Selección de ancho de ventana de tiempo 94
6.4 Diseño experimental 94

7 Entrenamiento de modelos 96
7.1 Creación de datastets de features 96
7.1.1 Metodología 97

7.1.2 Resultados y discusión 98
7.2 Creación de datastets de *labels* 98
7.2.1 Metodología 98
7.2.2 Resultado y discusión 99
7.3 Ejecución de experimento fase 1 99
7.3.0.1 Curvas de aprendizaje 102
7.3.1 Metodología 104
7.3.2 Resultados y discusión 104
7.4 Ejecución de experimento fase 2 108
7.4.1 Metodología 110
7.4.2 Resultados y discusión 110
7.5 Validación de modelos con datos de línea de tiempo completa 112
7.5.1 Metodología 112
7.5.2 Resultados y discusión 113
7.6 Filtrado mediante contador *up-down* 114
7.6.1 Metodología 115
7.6.2 Resultados y discusión 115
7.7 Aislamiento de termopar con fallo 118

Bibliografía **123**

Resumen

Detección y Aislamiento de Fallas en una Red de Termopares Mediante Técnicas Basadas en Datos

En el presente documento se explica el proceso de desarrollo de un modelo para detección y aislamiento de fallos (*FDI*) en una red de termopares mediante técnicas basadas en datos. El documento inicia describiendo generalidades del funcionamiento de la planta, conceptos básicos sobre termopares, definición de *FDI*, su relevancia en la planta y el modo en que este se desarrolla actualmente, lo cual abre paso a la identificación del problema y el planteamiento de los objetivos. El desarrollo del proyecto se divide en 4 etapas, iniciando con el reconocimiento del conjunto de datos disponibles, seguido de un estudio de métricas obtenidas a partir del conjunto de datos y su relación con estados de fallo o normalidad en los termopares, establecimiento de una metodología para el entrenamiento de modelos basados en datos y los resultados obtenidos de su aplicación. El documento finaliza con la determinación de parámetros para la construcción un modelo basado en datos que muestra una precisión superior al 76 %, según pruebas de validación aplicadas, entre otras conclusiones obtenidas del desarrollo del presente proyecto.

Palabras clave: Aprendizaje supervisado, *data-driven*, detección, fallas, *FDI*, *Machine Learning*, sensor, termopar, *XGBoost*.

Abstract

Fault detection and isolation in a thermocouple network by data-driven techniques

The present document explains the development process of a fault detection and isolation (FDI) model for a thermocouple network by data-driven techniques. The begins by describing plant's functioning generals, thermocouples' basic concepts, FDI definition, its relevance for the plant and how it is currently performed, which allows the problem's identification and objectives definition. The project's development divides into 4 stages, starting by a reconnaissance of available data, followed by a study of metrics obtained from the data set and their linkage to thermocouple's in fail or normality statuses, establishment of a methodology for training data-driven models and its results. The document finalizes by determining the parameters for the constructions of a data-driven model showing an accuracy over 76 %, according to applied validation tests, among other conclusions from the development of the present project.

Keywords: Supervised learning, data-driven, detection, failures, FDI, Machine Learning, sensor, thermocouple, XGBoost.

Agradecimientos

La materialización del presente trabajo no hubiera sido posible sin la ayuda de un sinnúmero de personas cuyos valiosos aportes marcaron la diferencia y fueron, cada uno, piedra angular en diferentes etapas del desarrollo de este proyecto.

A los profesores Jorge Sofrony y Erwin López, quienes cada semana me orientaron y motivaron a profundizar mi conocimiento, guiándome durante el desarrollo de este proyecto que inició con grandes expectativas y poca información sobre cómo llegar a destino para convertirse en la realización que hoy se plasma en este documento.

A ti, Amelia, quien con gallarda paciencia e infinito esmero me acompañaste y diste soporte cada día extenuante, cada noche insomne, cada eterna hora de escucharme hablar sobre un único tema... y uno que otro ataque de ansiedad... no me alcanzan las palabras para agradecerte lo suficiente.

A mis padres, de quienes aprendí que la constancia, el trabajo arduo y la educación son motor fundamental para crecer como profesionales y como personas de valor en la sociedad.

A mis colegas y amigos que pusieron su tiempo, conocimiento, apoyo y valiosos consejos todo este tiempo. A todos y cada uno les debo una parte de esta meta alcanzada. A todos les estaré agradecido cada día.

Por último, pero no menos importante, quisiera agradecer a mi *alma mater*, la Universidad Nacional de Colombia, por ser esa mamá tanto permisiva como exigente a quien le debo mi formación, de la que tengo más recuerdos de los que pudiera contar, de la cual me siento cada día más orgulloso por haberme formado en sus aulas y, que hoy en día recuerdo con creciente nostalgia desde este rincón del globo desde donde escribo este sentido agradecimiento. Gracias por hacerme quien soy.

Cada vez que nos equivocamos, estamos más cerca de la verdad.

1 Introducción

1.1. Estructura del documento

En el presente capítulo se realiza una presentación general de la planta, el proceso y el sistema en el que se identifica la oportunidad para desarrollar el proyecto plasmado a lo largo de este documento. Se explica el concepto del termopar desde un esquema básico de su construcción, se introducen el concepto y las condiciones de linealidad en un termopar y se plantea una primera noción del concepto de fallo en un termopar respecto a su modelo en la zona lineal. Posteriormente se introduce el concepto de detección y aislamiento de fallos y tres aproximaciones que se pueden emplear para llevar a cabo esta tarea. Se muestran aspectos generales y algunos desarrollos generados desde cada una de estas aproximaciones. La introducción se propone en este orden para contextualizar progresivamente al lector y empezar a establecer las conexiones e importancia de cada uno de estos componentes en el desarrollo del presente trabajo.

El documento se organiza en 7 capítulos. En el capítulo 3 se establecen el objetivo general y los objetivos específicos del presente trabajo. El desarrollo del proyecto se divide en cuatro macroprocesos, cada uno correspondiente con cada uno de los objetivos específicos:

1. Caracerización y preprocesamiento de datos iniciales.
2. Establecimiento de métricas y línea base.
3. Definición de diseño experimental.
4. Entrenamiento de modelos.

En cada capítulo se realiza una introducción a la idea que se desarrollará y se establece la metodología a seguir. Posteriormente se muestra el resultado del ejercicio desarrollado y se propone una breve discusión de los resultados obtenidos en el capítulo.

El capítulo 4 muestra el proceso desarrollado en la etapa de caracerización y preprocesamiento del conjunto de datos suministrado por CMSA. Esta etapa permite hacer un reconocimiento inicial de los datos disponibles y establecer un grupo de datos de interés para el desarrollo del presente trabajo.

El capítulo 5 ilustra los procesos de Análisis Exploratorio de Datos (*EDA*), exploración de métricas relacionadas con estados de falla en termopares y establecimiento de una línea base, lo cual hace posible entrenar modelos basados en datos.

Los capítulos 6 y 7 se articulan como etapas planeación y ejecución. Se parte de un diseño experimental para realizar un proceso de variación de parámetros cuyos resultados permiten refinar un diseño experimental posterior. Al final del capítulo 7 se define la configuración de parámetros que ofrece los mejores resultados bajo el alcance del presente proyecto. Queda abierta la posibilidad a que futuros trabajos sigan experimentando con diferentes arquitecturas, modelos, metodologías, y demás posibles variantes en busca de nuevos enfoques y resultados.

Las conclusiones finales se presentan en el capítulo 7.7 con lo que se da cierre al trabajo desarrollado y se sintetizan ideas globales producto tanto del proceso como de los resultados obtenidos.

1.2. La planta y el proceso

Cerro Matoso S.A. (CMSA) es una operación del Grupo Industrial *South32*, encargada de la mina de explotación a cielo abierto de ferroníquel ubicada en cercanías del municipio de Montelíbano, departamento de Córdoba, Colombia. La operación se compone de las etapas de extracción de mineral, trituración, clasificación, secado, calcinación, fundición, extracción de metal fundido, extración de escoria y empaque del producto obtenido.

El diagrama de la figura **1-1** muestra el proceso entre la etapa de secado del mineral extraído hasta la extracción tanto del metal fundido como de la escoria. El mineral extraído es clasificado, separado en pequeñas montañas denominadas pilas, secado y posteriormente pre-quemado en los calcinadores a una temperatura de aproximadamente 800 °C, de lo que se obtiene uno de los productos intermedios clave en el proceso, la calcina.

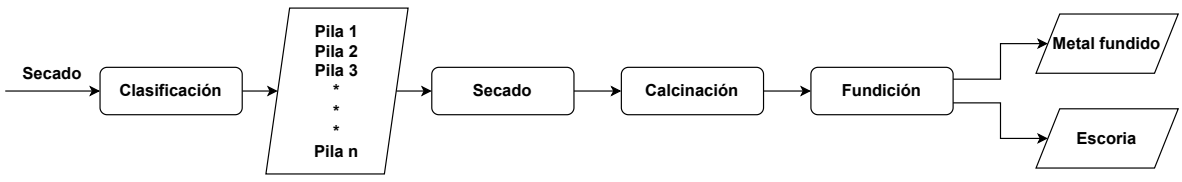


Figura 1-1: Diagrama de flujo (parcial) del proceso de obtención de ferroníquel

El proceso de fundición del mineral previamente triturado, secado, clasificado y calcinado se lleva a cabo en hornos de arco eléctrico que operan de manera ininterrumpida, salvo por paradas de mantenimiento programadas. La planta actualmente cuenta con dos hornos de fundición.

Los hornos reciben la calcina precalentada y se vierte desde la parte superior del horno por medio de tubos de alimentación que la depositan en diferentes zonas de acuerdo con los parámetros definidos en el procedimiento operativo. Los operarios ajustan parámetros como la alimentación de calcina y posición de los electrodos para buscar obtener el mayor volumen de producción de metal fundido y mantener el horno operando dentro de parámetros establecidos para reducir el desgaste que el material en reacción ejerce en las paredes del horno.

A medida que la calcina se calienta y pasa a fase líquida, esta se separa del metal que se deposita al fondo del horno y se extrae a través de piqueras ubicadas en la base de la estructura. Los componentes de la calcina que no constituyen el metal fundido se acumulan como escoria del proceso, la cual flota sobre el metal fundido y se extrae por medio de piqueras de escoria.

Se debe resaltar que la temperatura del material dentro del horno durante el proceso de fundición no es homogénea dado que dicha temperatura depende de factores como:

- Composición química de la calcina.
- Cantidad y distribución de la calcina.
- Distancia de la calcina respecto al electrodo.
- Profundidad del electrodo.
- Transferencia de energía del electrodo a la calcina.

A medida que el material se sitúa más lejos del electrodo, este recibe menos energía en forma de calor y, por ende, se calienta menos que el material más cercano al electrodo. Bajo ciertas condiciones de temperatura, se forma de una capa de material en fase sólida (material sin fundir) en la cara interna de la pared del horno [2]. Dicha capa, llamada costra, puede contener tanto metal sin fundir como minerales de la calcina, tiene dos características principales. Por una parte es material que no se extrae en forma de metal refinado y que, a su vez, ocupa espacio dentro del horno, reduciendo el volumen de producción; por otra parte, la capa de depositada en la pared del horno actúa como un recubrimiento protector que recibe los ataques térmico y químico del metal fundido, protegiendo la pared de ladrillo y prolongando la vida útil del horno.

La determinación del espesor del recubrimiento protector del horno es por si mismo un proceso que recubre especial relevancia en el control estructural de la planta. La medición directa del espesor de dicho recubrimiento requiere una parada total de la planta, proceso que requiere varios días para evacuar el horno, permitir el enfriamiento de la estructura, realizar la medidas y el posterior procesamiento de los datos recolectados, como lo ilustraron Tibaduiza *et al.* [3]. Los autores emplearon métodos analíticos para la medición del espesor del recubrimiento mediante ultrasonido en el horno detenido, resaltando las limitaciones que se presentan al requerir la parada de planta para obtener los datos y su imposibilidad de realizar las mediciones en línea.

Tibaduiza *et al.* [4] propusieron posteriormente un método para realizar la medición del recubrimiento sin requerir la parada de planta mediante la adaptación de un sistema *Ground-Penetrating Radar (GPR)* y el posicionamiento estratégico de antenas para recibir las señales recibidas por los emisores. El modelo propuesto requiere especial cuidado de las antenas de recepción debido a las altas temperaturas a las que se encontrarían expuestas al estar en contacto directo con la pared lateral del horno.

La planta cuenta actualmente con un sistema de control de alimentación de calcina y un sistema de monitoreo del estado estructural del horno, dichos sistemas guían al equipo operativo para garantizar la integridad estructural del horno durante la operación de fundición. El problema de estimación del espesor de costra y el desarrollo de métodos para controlar la formación de la costra son retos académicos que ameritan ser investigados en el futuro.

Además de las temperaturas del horno, se lleva un monitoreo constante de diferentes variables del proceso de producción, que incluyen volúmenes de mineral que ingresa para procesamiento, composición química de las diferentes pilas de mineral, temperaturas de calcinación, potencia eléctrica, corriente, impedancia y posición de los electrodos que calientan la calcina, flujo de escoria a la salida del horno, flujos de agua y flujos de calor retirado de cada panel, entre otros. Dicho monitoreo y almacenamiento de datos históricos permite hacer seguimiento de la operación de la planta.

1.2.1. Enfriamiento de las paredes laterales del horno

Una de las operaciones necesarias para el control estructural del horno es la extracción de calor de la pared media del horno. Esta operación se realiza mediante intercambiadores que extraen el calor de la pared a través de elementos metálicos ubicados en la parte externa de la pared del horno. Un flujo ininterrumpido de agua atraviesa los elementos metálicos y extrae el calor para ser disipado fuera de la línea de fundición.

Para monitorear el estado de operación de los hornos de fundición, se cuenta con una red de termopares dispuestos tanto en las paredes del horno como en diferentes puntos del techo y cara externa del fondo, conformando un conjunto de más de 200 termopares que operan continuamente.

La extracción de calor en la pared media del horno se realiza para evitar que éstas se deformen debido a las elevadas temperaturas del proceso, al tiempo que propicia la formación del recubrimiento protector al interior del horno. Los elementos o placas metálicas en la pared media del horno reciben los nombres de *Plate Coolers* y *Waffle Coolers*, los cuales se organizan verticalmente en 4 niveles de *Plate Coolers*, identificados con letras de la A a la D, y un nivel de *Waffle Coolers* [1]. La figura **1-2** muestra la ubicación relativa de los paneles que permiten el intercambio de calor.

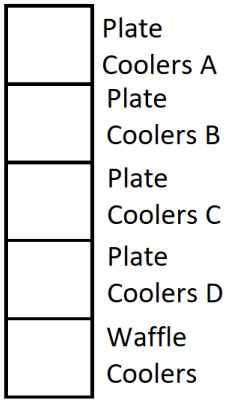


Figura 1-2: Ubicación relativa de *Plate Coolers* y *Waffle Coolers*

La circunferencia de la pared del horno se divide en 4 cuadrantes (noroeste, noreste, suroeste y sureste) y cada cuadrante a su vez se divide en 18 secciones, con lo que la totalidad del horno queda dividido en 72 secciones angulares como se muestra en la figura **1-3**. Dichas secciones angulares sirven para establecer la ubicación angular de los paneles a través de los cuales se mide la temperatura de la pared del horno.

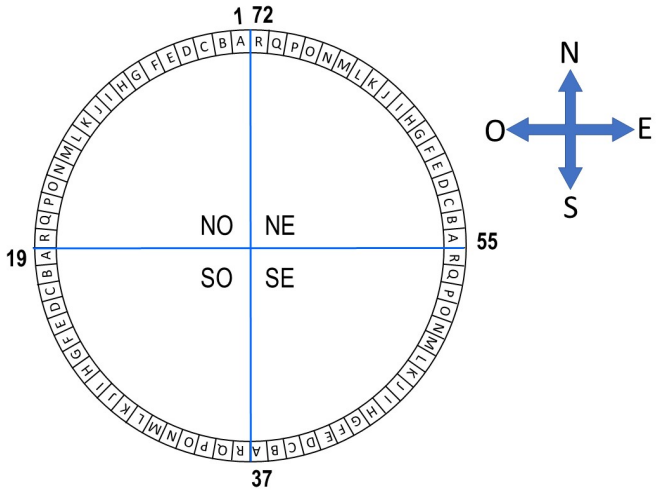


Figura 1-3: Distribución de secciones angulares del horno [1]

En los niveles A, B y C de *Plate Coolers* los paneles se ubican de forma escalonada, en el nivel D se ubica un panel en cada uno de los 72 sectores de la pared del horno, como se muestra en la figura **1-4**; el nivel de *Waffle Coolers* cuenta con un panel centrado en cada sector del horno y dos termopares (A y B) por cada panel.

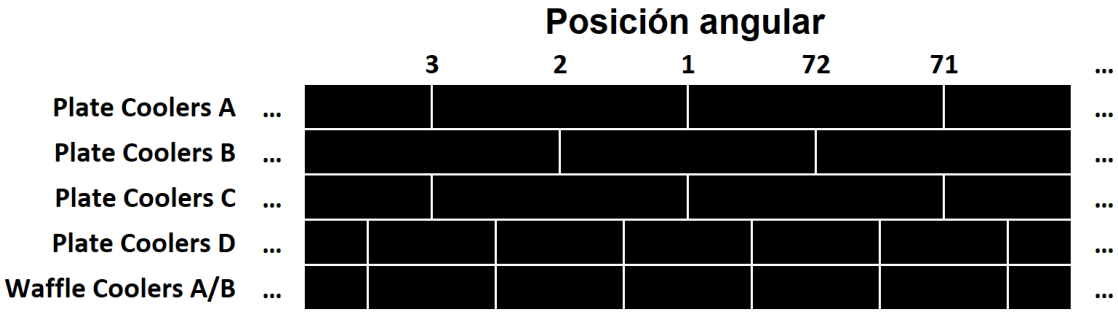


Figura 1-4: Ubicación de paneles en la pared del horno.

De la figura **1-4** también se observa que los *Plate Coolers* del nivel B se centran en las posiciones angulares impares, por lo que en la nomenclatura de los paneles no existen *Plate Coolers* del nivel B con posición angular impar. De manera análoga se observa que los *Plate Coolers* de los niveles A y C se centran solamente en las posiciones angulares pares, por lo que en la nomenclatura de los paneles solamente se encontrarán estos paneles acompañados de posiciones angulares pares.

La correcta operación del horno y su sistema de control necesita que las medidas que se obtienen de la red de sensores refleje fielmente los valores reales de temperatura en cada uno de los puntos de medida, por lo que resulta de especial importancia garantizar o, en la medida de lo posible, incrementar la confiabilidad de las medidas de los termopares para la operación del horno.

1.2.2. Detección e identificación de fallos en termopares del horno

Actualmente la detección de fallos en el correcto funcionamiento de los termopares del horno se realiza mediante inspección visual de los datos recibidos de los sensores por parte del personal operativo de planta, quienes revisan los registros históricos de los valores medidos en cada termopar y determinan si los datos corresponden a un sensor operando con normalidad o si, por el contrario, los datos reflejan que existen fallos en el sensor. Realizar dicho proceso para la totalidad de los termopares demanda grandes cantidades de tiempo y esfuerzo, además de requerir dedicación exclusiva si se quiere detectar fallos a poco tiempo de que estos suceden.

El uso de herramientas tecnológicas para apoyar el proceso de detección de fallos en la red de termopares se presenta como una estrategia atractiva para incrementar la confiabilidad en las mediciones, dada la cantidad de termopares, el ambiente extremo en que estos operan y su importancia para la correcta operación del horno.

En el intervalo de tiempo entre la aparición de una falla hasta su detección e identificación se corre el riesgo de tomar decisiones que desmejoren las condiciones de operación del horno, reduciendo la productividad o aumentando el desgaste de la pared. Dichas condiciones pueden conducir a efectos indeseados como el enfriamiento o sobrecalentamiento del horno, impactando negativamente en la eficiencia de la operación. En casos más drásticos, desviaciones en la operación del horno pueden conducir a incidentes graves tales como la perforación de la pared del horno [5], poniendo en riesgo la estructura, el personal y el entorno.

1.3. Termopares

Un termopar, como el esquematizado en la figura **1-5**, es un sensor compuesto por una juntura bimetálica que presenta una diferencia de potencial eléctrico, también denominada como tensión eléctrica, al someter los dos extremos de la juntura a una diferencia de temperatura, lo que se conoce como efecto Seebek [6].
La tensión eléctrica explicada mediante el efecto Seebek depende de las características fisicoquímicas de ambos metales, en lo que se conoce como coeficientes de Seebeck, y de la

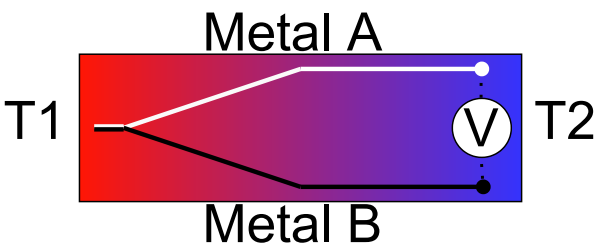


Figura 1-5: Esquema básico de un termopar sometido a una diferencia de temperatura.

diferencia de temperatura entre los dos extremos de la juntura, como se muestra en la ecuación (1).

$$V = \int_{T_1}^{T_2} (S_B(T) - S_A(T))dT \qquad (1)$$

Los coeficientes de Seebek $S_B(T)$ y $S_A(T)$ se presentan, generalmente, en unidades de $\mu V/K$ y su magnitud no es lineal respecto a todas las variaciones de temperatura. Sin embargo, existen combinaciones de junturas bimetálicas que, dentro de ciertos rangos de temperatura, presentan un comportamiento tal que la tasa de cambio de la tensión inducida por el fenómeno de Seebek y la diferencia de temperatura entre sus terminales puede considerarse lineal.

Normas técnicas, como la IEC 60584-1 [7], explican en detalle los requisitos técnicos que satisfacen termopares agrupados dentro de designaciones internacionales. Uno de los parámetros determinantes a la hora de seleccionar un termopar es el rango de temperatura de operación del sensor. Dicho rango nos permite trabajar bajo el modelo en que la medida de tensión en los terminales del termopar cambia de manera lineal respecto al cambio de temperatura en la punta caliente del sensor.

Mediante el uso de un transductor configurado de acuerdo con el tipo de termopar conectado y, si la punta caliente del termopar se encuentra dentro de cierto rango de operación específico para cada tipo de termopar, es posible establecer una relación lineal entre la tensión en los terminales del sensor y la temperatura a la que se encuentra expuesta la punta caliente, como se muestra en la ecuación (2), donde k es un factor lineal que depende del tipo de termopar.

$$V = kT \qquad (2)$$

Dado que el principio de medición de los termopares depende de la exposición (directa o indirecta) del sensor a la fuente de calor, los termopares en ambientes industriales suelen encontrarse expuestos a condiciones ambientales similares o iguales a aquellas en las que sucede el fenómeno térmico medido, lo que ocasiona una progresiva degradación o desgaste del sensor y genera que su comportamiento difiera cada vez más del ideal plasmado en la ecuación (2).

Una falla en un sensor implica una pérdida de función parcial o total de su capacidad de medida. Condiciones ambientales extremas de temperatura, acidez, ruido electromagnético, entre otras, aceleran el desgaste del termopar y propician la aparición de errores de precisión, estabilidad y confiabilidad [8]. Este desgaste se manifiesta como degradación en su funcionamiento, afectando las lecturas obtenidas. Las consecuencias del mal funcionamiento de un sensor se encuentran en diversas escalas, llegando en casos extremos a comprometer la seguridad e integridad tanto del personal como de la planta misma [9].

1.3.1. Clasificación de fallos en termopares

Con base en lo expuesto por Zhang *et al.* [9] y Samara *et al.* [10] respecto a la clasificación de fallas en un sensor, estas se pueden separar en abruptas e incipientes.
Una falla abrupta es una alteración súbita en el funcionamiento del sensor, haciendo que el mismo se vuelva inoperante durante la duración de la falla. Un caso común se presenta cuando la señal de salida del sensor se ancla en un cierto valor independientemente del valor real de la variable medida, como se muestra en la figura **1-6** donde la señal del *Plate Cooler* 31 D presenta un pico poco antes del marcador de fecha 2017-11-29, luego de lo cual la señal leída de dicho sensor se ancla en un valor pese a que las señales de los *Plate Coolers* 30 D y 32 D (paneles vecinos) presentan variaciones propias de la operación normal de la planta.

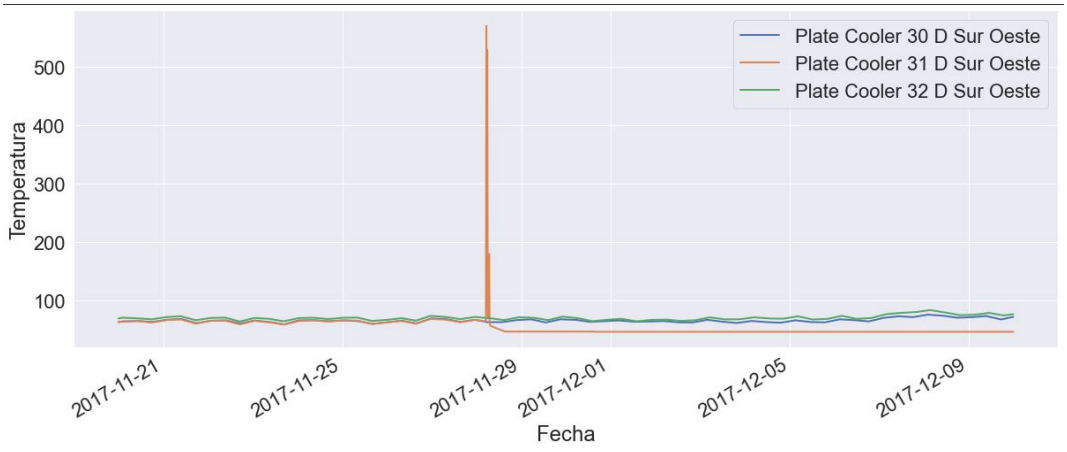


Figura 1-6: Ejemplo de sensor con señal anclada.

Algunos factores que pueden promover la aparición de fallas abruptas en termopares son cortocircuitos, desconexión (circuito abierto), daños físicos en el cableado, los terminales, entre otras.

Dentro de los fallos abruptos se encuentran los fallos intermitentes. Son eventos que no afectan el sensor de manera permanente, sino que suceden solamente en intervalos de tiempo, alternando intervalos de funcionamiento con fallos e intervalos de funcionamiento libres de fallos. Estos fallos, cuando no se deben a factores humanos tales como una conexión inestable o una desconexión accidental, suelen evolucionar con el tiempo hasta convertirse en un fallo permanente, como el ejemplo de la figura **1-7** en la que se observa que el *Plate Cooler* 36 A presenta un fallo transitorio entre las marcas de fecha 2015-07-15 y 2015-07-17; a partir del 2015-07-19 estos fallos transitorios se hacen cada vez más frecuentes hasta que entre el 2015-07-23 y el 2015-07-25 el sensor se ancla en un valor cercano a los 60 °C.

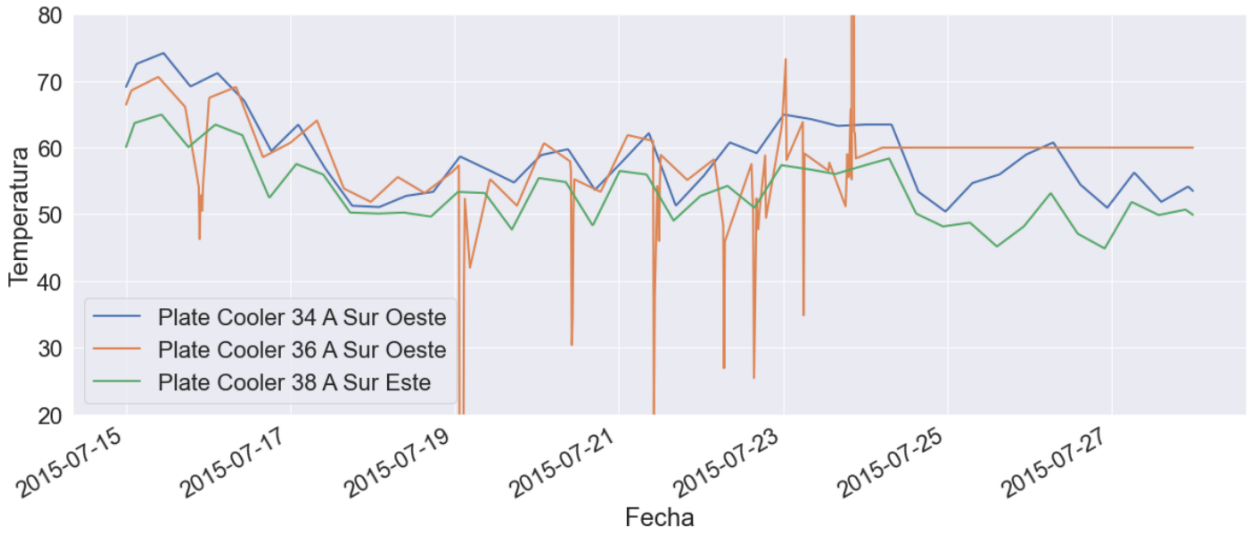


Figura 1-7: Ejemplo de evolución de sensor con fallo intermitente.

Los fallos abruptos pueden detectarse con relativa facilidad mediante observación directa ya que la señal entregada por el sensor en fallo presenta una marcada diferencia respecto al comportamiento de la variable medida en un sensor sin fallo. Esta identificación se puede realizar mediante comparación con señales redundantes, modelos matemáticos del sistema (si se dispone de tal modelo), indicadores derivados del conocimiento experto del sistema, entre otros métodos.

En un fallo incipiente el funcionamiento del sensor se ve modificado ligeramente sin que este deje de ser operativo, ocasionando que el dato entregado no sea veraz [11]. Esta clase de fallos suelen empezar como variaciones mínimas en el valor entregado, de modo que no son

fácilmente identificables [12]. Un fallo incipiente puede evolucionar hasta que la anomalía en el sensor sea tal que se dispare una falla abrupta. Sin embargo, el tiempo y velocidad de la evolución del fallo no son predecibles, lo que dificulta el establecimiento del punto de inicio de tales fallos una vez son detectados.

En un sistema de monitoreo de tipo industrial intervienen también sistemas como conversores análogo-digital (*ADCs*), *dataloggers*, *datacenters*, etc. En el presente documento no se analizan fallos ocurridos en estos sistemas, debido a que el estudio se enfoca exclusivamente en fallos en termopares.

1.4. Detección y Aislamiento de Fallos (FDI)

La Detección y Aislamiento de fallos (*FDI* por sus siglas en inglés) es un campo dentro de la ingeniería de control que cobra cada vez mayor importancia a medida que los procesos son cada vez más diversos y se monitorean simultáneamente una mayor cantidad de variables. A medida que los sistemas se hacen más complejos y especializados, también lo ha hecho el diagnóstico de fallas.

Autores como J. Patton [13] y Shahnazi [14] dividen el proceso de *FDI* en cuatro etapas:

- **Detección**: Determinar si existe de una falla.
- **Aislamiento**: Determinar el origen de la falla.
- **Estimación**: Determinar si el sistema puede seguir en funcionamiento.
- **Ajuste**: Modificar el sistema, bien sea para trabajar considerando la alteración en el sensor, o el cambio del elemento defectuoso

En la fase de detección se evalúa la existencia de anomalías en los datos del sensor que sugieran la existencia de una falla en el funcionamiento del mismo. La detección de fallos se basa en extraer información de los datos de entrada y, mediante uno o varios métodos, determinar la existencia de anomalías a partir de métricas de comparación conocidas como *residuos*. La estrategia de cómputo e interpretación de los residuos depende de la metodología de detección que se haya implementado.

Los residuos se pueden calcular mediante operaciones directas entre las variables de interés (lineales, polinomiales, etc.), transformaciones matemáticas (Fourier, Wavelet, Z, etc.), cálculos estadísticos, aplicación de modelos complejos (*Machine Learning*, inteligencia artificial, redes neuronales, etc.) o combinaciones que permitan aplicar un criterio de decisión al final del proceso para clasificar los datos en correctos o anómalos.

En la fase de aislamiento se toman los datos identificados como anómalos y, a partir del cómputo y evaluación de residuos, se decide el tipo de falla presente en los datos analizados. El modo en que las fallas se clasifican depende del contexto en que se esté trabajando, algunos posibles criterios para clasificar el origen de una falla son: el lugar físico donde ocurre, tipo de falla, consecuencia de la falla en el proceso, etc.

Una vez detectado y aislado el fallo, si se determina que el sensor puede seguir operativo, puede implementarse una cuarta etapa denominada ajuste en la cual se determina una estrategia para corregir los datos anómalos detectados y aislados. Esta etapa no siempre es realizable ya que depende de la capacidad para generar e implementar estrategias de recuperación o ajuste de datos.

Un modo de detección y diagnóstico de fallos en sensores es la redundancia física, que consiste en instalar dos o más sensores midiendo directa o indirectamente una misma variable en simultáneo. La comparación de las medidas de los diferentes sensores permite identificar si alguno de ellos registra medidas anómalas.

La redundancia física como método para *FDI* presenta una limitante fundamental a la hora de implementarse. Cada nuevo instrumento que se añade al sistema implica un despliegue de infraestructura y un costo asociado a dicho despliegue. Estos dos factores se hacen más críticos a medida que la complejidad del sistema incrementa, lo que se convierte en una limitante tanto tecnológica como financiera.

Existen diversas alternativas a la redundancia física para realizar *FDI*, especialmente en sistemas en que la cantidad de variables medidas o la complejidad del sistema hace imposible una comparación directa. Para abordar casos como este, existen tres grandes grupos de aproximaciones de *FDI* con redundancia analítica:

- Basadas en modelo
- Basadas en conocimiento
- Basadas en datos

La existencia de diferentes aproximaciones de *FDI*, empezando desde los sistemas basados en modelos derivados del trabajo de Willsky y Jones en 1976 [15], hasta modelos modernos que integran múltiples modelos de *Machine Learning*, técnicas de tratamiento de datos en diversas etapas del proceso, transformaciones y descomposiciones complejas de la información para resolver problemas de elevada complejidad y abstracción, no han generado un obligado desplazamiento de una u otra aproximación de *FDI*, sino que han permitido ampliar cada vez más la gama de posibles metodologías que se pueden adoptar para solucionar un problema

de detección de fallos en sistemas.

La inclinación por una u otra aproximación de *FDI* depende de diversas características y limitaciones del caso de estudio. Factores como:

- Complejidad (no linealidad) de la planta y/o el proceso.
- Disponibilidad, calidad y cantidad de datos de la línea base de conocimiento.
- Existencia y tipo de relaciones entre la línea base y los resultados esperados.
- Requerimientos específicos propios del caso de estudio.

Permiten descartar o apoyar la adopción de una aproximación de FDI para un caso de estudio en específico.

1.5. Aproximaciones Basadas en Modelos

Esta aproximación depende de la disponibilidad de un modelo matemático del sensor y del proceso que permita obtener analíticamente el comportamiento esperado de las mediciones y compararlo con los datos medidos, como se muestra en la figura **1-8**. En esta aproximación el residuo que se somete a los criterios de decisión es la diferencia entre el valor esperado y el valor medido del proceso. Esta aproximación se ve limitada en la medida que los procesos se hacen cada vez más complejos, dificultando la obtención del modelo matemático correspondiente para realizar el cálculo de los residuos.

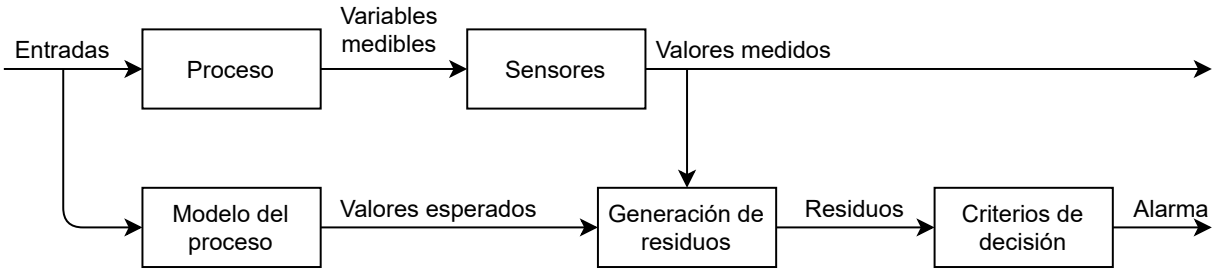


Figura 1-8: Diagrama de flujo de aproximaciones basadas en modelo

Las aproximaciones basadas en modelos son de gran utilidad en procesos cuyas variables de interés guardan relaciones fácilmente modelables y se tiene información de entrada suficiente, con lo que se pueden calcular los valores esperados y compararlos con las mediciones para la generación de residuos y determinación de las reglas de decisión.

La redundancia analítica permite reducir en algunos casos la cantidad de sensores redundantes [16], aprovechando los modelos matemáticos de los sistemas y obteneniendo información acerca de la señal de interés a través de combinaciones de las señales disponibles. La figura **1-9** presenta la diferencia entre redundancia física y redundancia analítica a partir de un sensor inicial dentro de un proceso.

Los métodos basados en modelos analizan las entradas, estados y/o salidas del sistema y las comparan con las mediciones registradas para el cálculo de residuos. Una de las primeras nociones de este método se desarrolló a partir del trabajo de Alan Willsky en 1976 [15], donde se establece la señal residual como la diferencia entre los valores calculados en el modelo y los valores medidos en el proceso para detectar desviaciones entre el comportamiento esperado de sistema y el comportamiento observado. El comportamiento de esta señal residual, combinado con el establecimiento de umbrales de operación y reglas de toma de decisión ha permitido definir diversos sistemas de diagnóstico de fallos.

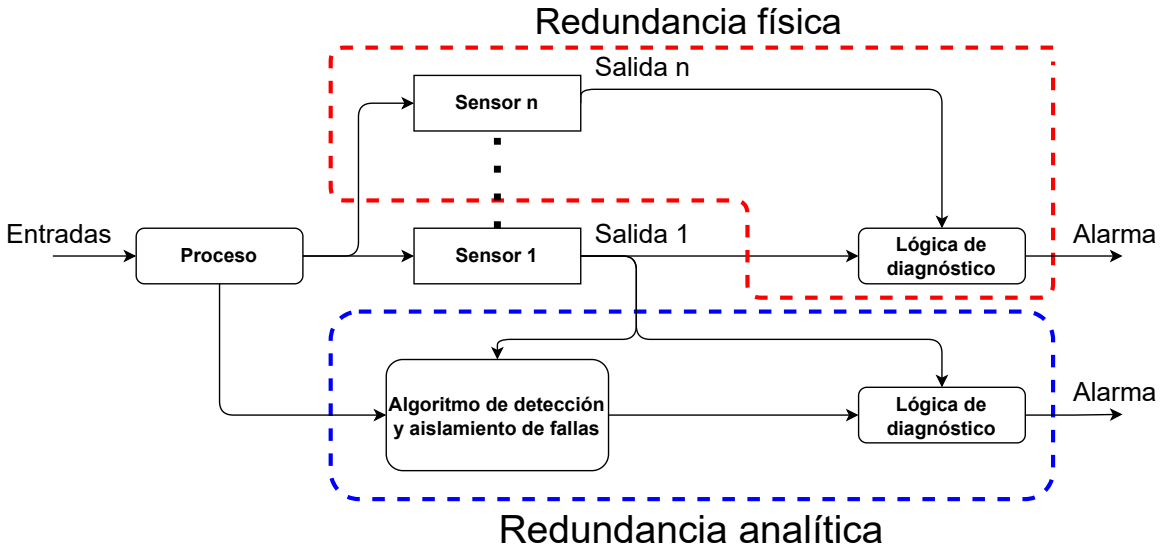


Figura 1-9: Redundancia física y redundancia analítica

El establecimiento de la función residuo y los procesos de discriminación y toma de decisiones dependen de cada caso de estudio particular, como los presentados por Tolouei y Shoorehdeli [17] donde se establece un sistema de observadores no lineales como una alternativa para establecer redundancia analítica para un sensor de medición de pH en un sistema de neutralización y, de esta manera, detectar fallos a partir del residuo calculado con base en la salida del observador y el valor entregado por el sensor. Para el establecimiento del observador del sensor de pH emplearon datos obtenidos de otros sensores de la planta, como niveles y flujos de los tanques de cada reactivo de la neutralización, además de los valores característicos de acidez de los reactivos que ingresaban al reactor.

Cheng *et. al* [18] establecieron un modelo de *FDI* para la detección de fallas por cortocircuito y fallos en los sensores para el sistema de baterías de ion de litio de una aeronave con base en un análisis estructural a partir de la combinación de modelos eléctricos. Los autores establecieron modelos para cada celda, para cada módulo (conjunto de celdas en serie), para el conjunto de celdas en paralelo y para el circuito equivalente visto desde la carga. Con base en la caracterización de la condición inicial de las celdas, los módulos y el circuito equivalente. Los autores descomponen el modelo en sus múltiples componentes, comparan las medidas obtenidas de sensores de tensión en cada celda y de corriente tanto en cada módulo como en la carga y comparan los datos contra los resultados calculados para el sistema caracterizado en su condición inicial para detectar la existencia de un fallo y aislar la fuente del fallo.

Shahnazari *et al.* [19] desarrollaron un sistema de *FDI* para un sistema de acondicionamiento de aire, definieron el sistema a partir de un modelo lineal invariante en el tiempo y diseñan observadores de estados para estimar los residuos con base en los valores de variables de estado. Los autores logran detectar efectivamente la aparición de fallos simulados en el modelo.

Las aproximaciones basadas en modelos dependen de la calidad de la línea base que se tenga del caso de estudio. En los casos en que el modelo existente es suficientemente preciso, los ejercicios de *FDI* suelen ser bastante efectivos e intuitivos en su interpretación ya que se puede establecer una relación directa entre el residuo y la existencia fallos.

Por otra parte, cuando los modelos no son lo suficientemente precisos, pueden aparecer errores considerables entre el comportamiento descrito por el modelo y el reflejado por las mediciones. Estos errores no necesariamente implican anomalías en el funcionamiento del sistema y se originan por falencias en la precisión del modelo utilizado [20]. Algunas condiciones que pueden reducir la precisión de un modelo incluyen, pero no se limitan a:

- **Sobresimplificación del modelo:** Ocurre cuando, a fin de darle al modelo mayor simplicidad matemática, se desprecian o alteran en exceso dinámicas del modelo real, lo cual desencadena que el comportamiento del modelo simplificado difiera notablemente respecto al modelo real.
- **Selección errada de parámetros:** Generalmente los modelos requieren del conocimiento de valores de condiciones iniciales del modelo; constantes físicas, dimensiones, condiciones ambientales, entre otros, ejemplifican algunos tipos de parámetros que pueden ser requeridos para el uso de modelos. Una errada selección de parámetros ocasiona que, a pesar de usar el tipo correcto de modelo, el modelo no refleje el comportamiento real del sistema.
- **Factores no modelados:** Correspondientes con características y condiciones del modelo que no fueron consideradas dentro de la definición de los modelos y que, ines-

peradamente, afectan el comportamiento medido del sistema: influencia de agentes externos, cambios internos en el sistema debidos a factores no modelados (aleatorios o determinísticos), condiciones iniciales desconocidas, etc.

Dadas las dimensiones, modo de operación, la constante intervención de los operarios en el monitoreo y ajuste de la operación del sistema, cambios no predecibles en la calcina a la entrada del horno, desconocimiento de algunos parámetros necesarios para el establecimiento de modelos detallado del horno y de las reacciones físicas, químicas, eléctricas y térmicas que intervienen en el proceso productivo, no es posible considerar la exploración de estrategias de *FDI* basada en modelos para la detección y aislamiento de fallos en termopares del horno.

1.6. Aproximaciones Basadas en Conocimiento

Los métodos basados en el conocimiento no requieren del modelo matemático descriptivo del sistema a estudiar, sino que analizan la información histórica de fallos previamente identificados y la combinan con conocimiento experto mediante un motor de inferencia difusa, como en el modelo mostrado en la figura **1-10**, para determinar la existencia y clasificación de nuevos fallos, permitiendo un mejor diagnósticos de las fallas a medida que el sistema aprende de las entradas que recibe.[21] [22]

Dado que no se basa en un modelamiento matemático descriptivo del proceso, permite un diagnóstico más simple en sistemas no lineales.

El motor de inferencia recibe los valores medidos y, a partir de una base de conocimiento y una base de reglas determinadas por expertos, realiza el razonamiento que es leído por un intérprete para generar los residuos que serán sometidos a los criterios de decisión internos. Las alarmas generadas son presentadas a los usuarios para que éstos indiquen al sistema de lógica difusa si la alarma corresponde a una falla real.

Una de las aplicaciones tempranas de la lógica difusa en la toma de decisiones fue expuesta por Zadeh en 1982 [23] donde se establecieron tratamientos para la falta de precisión y eventos difusos. A partir de este referente se han generado diferentes desarrollos para dar solución a problemas en sistemas de control. El sistema aprende y mejora con base en la realimentación del usuario a la base de conocimiento y las modificaciones en la base de reglas por parte de los expertos. Esto hace que su planteamiento inicial, así como su mantenimiento, dependan en gran medida del conocimiento y la interacción con los expertos, haciéndolo altamente sensible a la calidad de los aportes generados por los expertos.

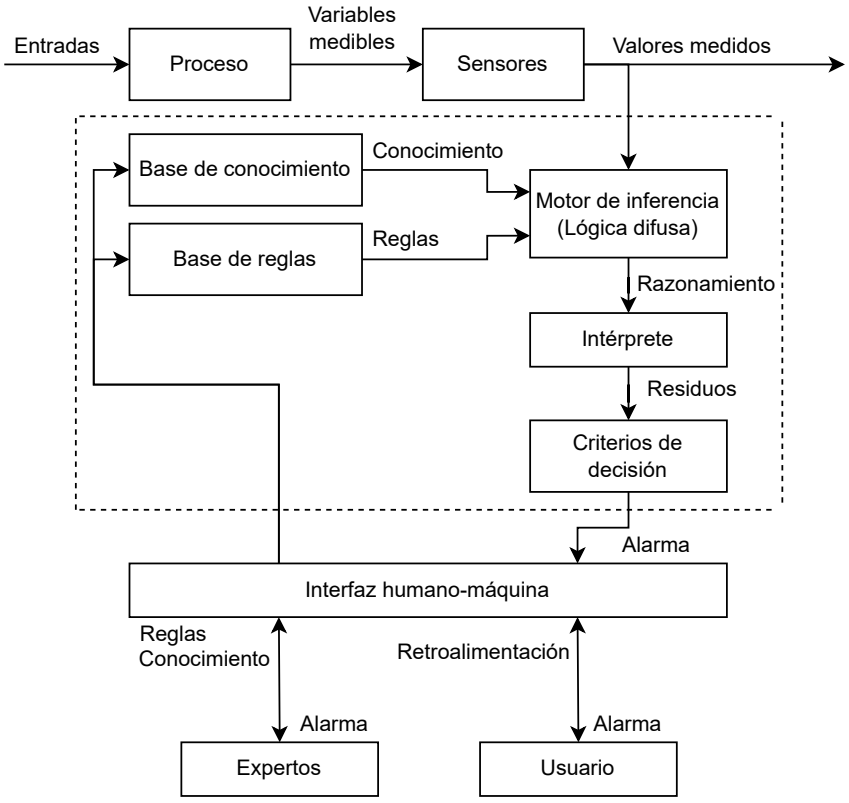


Figura 1-10: Diagrama de flujo de aproximaciones basadas en conocimiento

En 2018, Raja *et. al* [24] emplearon lógica difusa para la detección y clasificación de fallas en nodos de sensores distribuídos espacialmente. Estos nodos no son necesariamente homogéneos y pueden estar sometidos a condiciones ambientales adversas. Fallas externas tales como agotamiento de la batería, fallas en la red de comunicaciones, entre otras, son condiciones posibles en esta clase de escenarios. Por tal razón los autores crean una estrategia basada en lógica difusa y un esquema de gestión de recursos donde los diferentes nodos son clasificados de acuerdo con sus funciones y se emplea la comunicación con nodos vecinos para determinar el estado y potencial falla (microcontrolador, batería, comunicaciones, etc.). Con base en la información recolectada, el sistema determina la mejor acción de ajuste para el funcionamiento del sistema distribuído.

En 2018, He et al. [25] utilizaron la metodología de inferencia BRB (*Belief-Ruled-Based*) [26] para describir el diagnóstico de falla del sensor en términos de las características del sensor como parte de una red distribuída y posteriormente se describe la falla en términos de tiempo, causas y atributos. Para incrementar la precisión del modelo, se emplea un algoritmo de adaptación de matriz de covarianza para optimizar los parámetros iniciales del modelo propuesto.

En 2019, Yuan et al. [27] combinaron la minería de reglas de asociación y la inteligencia artificial para la selección de reglas en sistemas autónomos de parqueo. A partir del modelamiento de datos del proceso de estacionamiento de vehículos, se asignaron atributos difusos simbólicos con base en las características de cada proceso de parqueo, similar al proceso cognitivo que sucede con las personas, donde las distancias y pasos no se miden numéricamente sino de manera cualitativa. Una vez recolectada la información de múltiples ejercicios, se establecieron criterios de tolerancia entre los requisitos de cada caso y las acciones necesarias para cumplir cada tarea del proceso de estacionamiento. El algoritmo compara los datos de los diferentes ejercicios y establece reglas para el estacionamiento autónomo.

En el año 2019, Maciol *et. al* [28] patentaron un algoritmo para minimizar el costo de la adquisición de datos para la conformación de la base de conocimiento del sistema basado en reglas. El trabajo realizado permite evaluar *a priori* el costo de adquisición de datos para un sistema, lo cual constituye una herramienta estratégica desde el punto de vista de inteligencia de negocios. El objetivo del algoritmo es reducir los costos asociados al personal especializado requerido para la construcción de la base de conocimiento.

La base de conocimiento de un sistema basado en reglas es constantemente modificada por la persona con quien este interactúa, de ahí la importancia que sea un experto quien genere la base de conocimiento. Sin embargo, es difícil establecer reglas para sistemas de alta complejidad dado que una falla particular podría tener múltiples causas y múltiples implicaciones, dificultando el establecimiento de reglas claras de decisión.

1.7. Aproximaciones Basadas en Datos (Data-Driven)

Las aproximaciones basadas en datos se cimientan en los modelos de *Machine Learning* y redes neuronales (*Neural Networks*). Resultan especialmente útiles cuando los modelos de los sistemas suponen una carga computacional elevada o el modelo es demasiado complejo para ser definido de manera analítica.

A diferencia de aproximaciones basadas en conocimiento, las aproximaciones basadas en datos se diferencian en que estas no dependen del establecimiento y mantenimiento de una base de reglas de decisión, por lo que se reduce la dependencia constante de un experto en el mantenimiento de dichas bases. Un modelo *Data-Driven* puede entrenarse una única vez y, si el entrenamiento se realiza apropiadamente, el modelo puede operar de manera autónoma hasta que ocurra un cambio sustancial en la planta. Los modelos de *Machine Learning* se clasifican en supervisados y no supervisados, a continuación se realiza una breve introducción de estos dos tipos de modelo.

1.7.1. Aprendizaje no supervisado

Los modelos no supervisados se basan en la reducción de dimensionalidad de la información de entrada y, con base en métricas, establecen grupos o *clusters*[1] de forma que los elementos dentro de cada cluster sean lo más similares entre sí y cada *cluster* sea lo más diferente posible a los demás.

Uno de los modelos más comunes de *Machine Learning* no supervisado es el *k-means* [29], que se basa en medir iterativamente la distancia Euclidiana entre los elementos a clasificar y los centroides de los k *clusters* hasta encontrar la agrupación con la mínima distancia entre los miembros de un mismo *cluster*, conservando la máxima distancia entre los centroides de cada cluster. La figura **1-11** muestra un ejemplo del uso del modelo *k-means* en un conjunto de datos generado aleatoriamente.

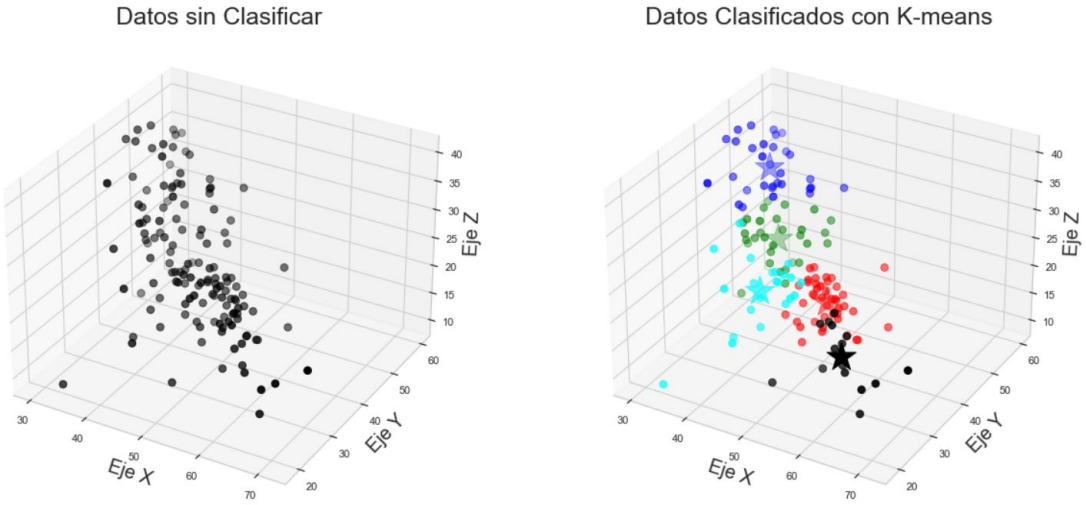


Figura 1-11: Ejemplo de clustering usando *k-means*

1.7.2. Aprendizaje supervisado

Los modelos de *Machine Learning* supervisados se basan en el uso de un modelo estadístico que produce una o varias salidas denominadas etiquetas o *labels* a partir de métricas de entrada denominadas *features* o parámetros, como se muestra en la figura **1-12**.

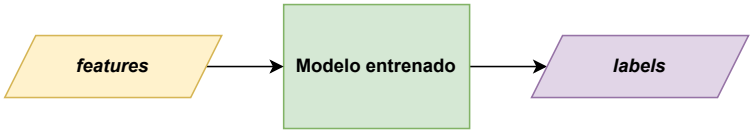


Figura 1-12: Diagrama de flujo de un modelo de *Machine Learning*

[1] Conjunto de observaciones o datos similares entre sí e identificados con una misma etiqueta.

Los modelos de *Machine Learning* supervisados pueden desarrollar tareas de regresión o clasificación. En las tareas de regresión se busca predecir el valor de una o más variables de salida (*labels*) a partir de los *features* de entrada, mientras que en las tareas de clasificación se busca establecer la pertenencia de una observación (caracterizada por su correspondiente conjunto de *features*) a un número finito de categorías definidas durante el entrenamiento del modelo, entendiendo que se definen dos o más posibles categorías.

La cantidad de categorías en un modelo de clasificación se establece en función de la tarea a desarrollar. Los modelos de clasificación pueden aplicarse en una diversa gama de campos y con múltiples propósitos, como se muestra en los ejemplos listados a continuación:

- Clasificación de hongos entre comestibles y venenosos con base en características fisionómicas y del entorno en que se encuentran. [30]
- Clasificación de tipos de accidentes cerebrovasculares a partir de imágenes de tomografía computarizada. [31]
- Detección de sitios web de *phishing* con base en el contenido de la página web e información del navegador. [32]
- Clasificación de poemas con base en características del lenguaje utilizado en el texto. [33]
- Clasificación de tipos fallos por deriva en sensores a partir de métricas derivadas de la señal entregada por el sensor. [34]

Los ejemplos listados previamente no abarcan la totalidad de campos y aplicaciones de los modelos de aprendizaje supervisado de *Machine Learning*. Estos ejemplos permiten hacerse una idea de la variedad de diferentes métricas que pueden emplearse como *features* para entrenar un modelo de *Machine Learning*. Sin embargo, no toda métrica es apropiada para el entrenamiento de un modelo, así como tampoco se puede garantizar que todo modelo entrenado con *features* derivados de las observaciones de entrada va a clasificar acertadamente dichas observaciones.

El proceso a seguir para obtener un modelo supervisado de *Machine Learning* se presenta en la figura **1-13**.

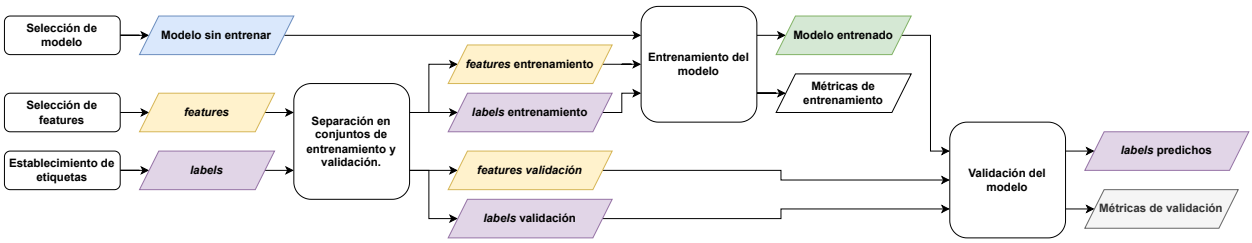


Figura 1-13: Diagrama de flujo para la obtención de un modelo de *Machine Learning* de aprendizaje supervisado

En la parte izquierda de la figura **1-13** se muestran 3 subprocesos que se articulan para poder obtener los 3 elementos principales del proceso, el modelo sin entrenar (azul), los *features* (amarillo) y los *labels* (púrpura).

1.7.2.1. Establecimiento de labels y features

Los *labels* pueden establecerse manualmente con base en el conocimiento previo que se tenga sobre el conjunto de datos (*dataset*), emplear criterios heurísticos o modelos de aprendizaje no supervisado, entre otros. La asociación de *labels* y observaciones es clave para el establecimiento de una línea base que de soporte al entrenamiento de modelos. En la línea base se incluyen, además de las observaciones y los *labels*, las relaciones de las variables y sus transformaciones con las categorías establecidas, aproximaciones heurísticas a soluciones al problema a resolver, resultados previos de aproximaciones a una solución y toda información que permita establecer un marco de referencia contra el cual comparar cualquier solución que se desarrolle.

Los *labels* deben ser establecidos para los modelos de clasificación, ya que en los modelos de regresión la salida del modelo es la o las variables cuyo valor se desea predecir.

La generación manual de etiquetas para una línea base requiere de una persona capacitada en el caso de estudio para juzgar cada observación y determinar la etiqueta que corresponde en cada caso. Una vez establecida una línea base, este conjunto de observaciones y etiquetas puede ser utilizado para evaluar tantas aproximaciones a una solución como se desee, siempre y cuando se garantice no modificar la línea base.

Una vez establecidas las etiquetas, realiza un Análisis Exploratorio de Datos (*EDA* por sus siglas en inglés) para caracterizar las variables que componen el conjunto de datos e identificar relaciones entre las diferentes variables de entrada y salida del caso de estudio. En un *EDA* se pueden emplear herramientas heurísticas, estadísticas, agrupaciones, visualizaciones, transformaciones, entre otras que se consideren oportunas para entender los datos e

identificar métricas de interés.

Existen casos en los que no se tienen previamente establecidas las categorías a utilizar para clasificar las observaciones. En estos casos resulta de especial interés apoyarse en métodos no supervisados, reglas heurísticas, métodos estadísticos, etc, para establecer categorías que se puedan emplear para etiquetar el conjunto de datos. En estos casos se debe realizar con mayor detenimiento el *EDA* para establecer las características, similitudes y diferencias entre las diferentes categorías y anticipar posibles problemas que pueda enfrentar el modelo de clasificación. Dentro de los problemas más comunes en esta clase de escenarios se puede mencionar la dificultad de realizar una clasificación acertada si dos o más clases presentan pocas diferencias entre sí.

Dependiendo del modelo a entrenar, puede ser necesario realizar transformaciones a los *features* para garantizar que el modelo puede operar los parámetros seleccionados. Algunas transformaciones comunes en el establecimiento de features incluyen, pero no se limitan a:

- **Normalización:** Transformación en la que se emplean los valores máximo, mínimo, media y varianza de un conjunto de datos conocidos para transformar dicho conjunto de forma que el resultado transformado se encuentre dentro de un rango deseado, normalmente $[0, 1]$ o $[-1, 1]$, junto a otros efectos estadísticos deseables como la unificación de las medidas de media y varianza del conjunto de datos.
- **Codificación:** Transformación en la que a cada uno de los n posibles valores de una variable categórica se le reemplaza por un número equivalente. La variable transformada será una variable numérica con n valores posibles que corresponden en posición con las categorías originales.
- ***One-hot encoding*:** Transformación empleada para el tratamiento de variables categóricas, en la que una única variable categórica con un número finito n de posibles valores es reemplazada por un número n de variables discretas booleanas mutuamente excluyentes, una variable *one-hot* por cada uno de los posibles valores de la variable original.
- **Eliminación de valor medio:** Tranformación aplicable a variables numéricas, en las que a cada valor del conjunto de datos se le resta el valor medio del conjunto de datos. Esta transformación se emplea con líneas de tiempo y es altamente sensible a la existencia de valores extremos que puedan introducir sesgo en la media.

Uno de los objetivos de estas transformaciones es reducir el riesgo de que el modelo entrenado genere sesgos en los parámetros de entrenamiento que puedan desbalancear la incidencia de los *features* en el cálculo de la variable de salida.

1.7.2.2. Selección del modelo

A partir de un *EDA* riguroso se pueden establecer variables, agrupaciones, transformaciones, entre otros, que muestren relaciones lógicas con las categorías o el valor de las variables de salida del caso de estudio. Dependiendo de las características del caso se estudio se puede seleccionar un grupo de modelos de *Machine Learning* que se ajustan mejor al tipo de tarea a desarrollar, así como se pueden descartar aquellos modelos que no son adecuados para el caso de estudio que se esté analizando.

En general, los modelos de aprendizaje supervisado de *Machine Learning* se basan en el uso de relaciones matemáticas, estadísticas, probabilísticas, transformaciones, reglas de decisión, etc. para estimar el valor de la variable dependiente con base en los *features*. Cada modelo tiene su propio modo de manipular las entradas para estimar salidas, por lo que, independientemente del modelo, la obtención de un modelo entrenado apropiado para la tarea establecida (regresión o clasificación) depende de un acertado proceso de selección de modelo y *features*.

En el ámbito de *Machine Learning* no suele haber un único modelo viable para realizar una tarea. Es común encontrar casos en los que existe más de un modelo con capacidad de desarrollar la tarea establecida; una aproximación para seleccionar el mejor modelo es entrenar los diferentes modelos considerados elegibles y establecer una o más métricas para comparar y determinar el modelo cuyos resultados satisfacen en mayor medida los objetivos del caso de estudio. Algunos modelos de *Machine Learning* se mencionan a continuación. [35]

Regresor lineal: Es un modelo simple de regresor que basa en la existencia de una relación lineal entre los *features* y la salida, como se muestra en la ecuación (3).

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + ... + \beta_n X_n + \epsilon \qquad (3)$$

Donde:

- Y: Variable dependiente o variable de salida
- X_i: Variables independientes, regresoras o *features* del modelo.
- β_i: Parámetros del modelo.
- ϵ: ajuste error y agrupa diferentes factores cuya influencia en el modelo no se relaciona directamente con las variables regresoras.

El regresor lineal requiere de la existencia de correlación entre las variables regresoras y la variable dependiente. De otro modo, el modelo no podrá estimar correctamente la variable dependiente.

Clasificador Bayesiano: Modelo de clasificador que se basa en la noción de que los atributos que determinan la categoría a la que pertenece una observación son mutuamente independientes, de forma que el modelo de probabilidad de pertenencia de una $\mathbf{x}$ observación a una i-ésima clase se puede expresar como el producto de las probabilidades condicionales de los n atributos que la caracterizan, siguiendo el modelo mostrado en la ecuación (4).

$$P(\mathbf{x}|c_i) = \prod_{j=1}^{n} P(x_j|c_i) \qquad (4)$$

El clasificador compara la probabilidad de que la muestra $\mathbf{x}$ pertenezca a cada una de las clases y la etiqueta con la categoría para la que haya obtenido la mayor probabilidad.

Este modelo requiere de pocas muestras para su entrenamiento y es robusto ante datos faltantes. Sin embargo, suelen ser inexactos debido a la suposición de la mutua independencia de los atributos, por lo que se le suele llamar clasificador ingenuo (*Naive Bayes Classifier*).

Árboles de decisión: Este modelo de clasificador se basa en el establecimiento reglas de decisión para cada parámetro del modelo de clasificación hasta la etiqueta que se considera más probable. Un árbol de decisión se puede ver como en el esquema de la figura **1-14**, donde cada bloque de decisión se tiene un criterio diferente y cada decisión puede llevar a un nuevo bloque de decisión o a una clasificación, representada como una hoja.

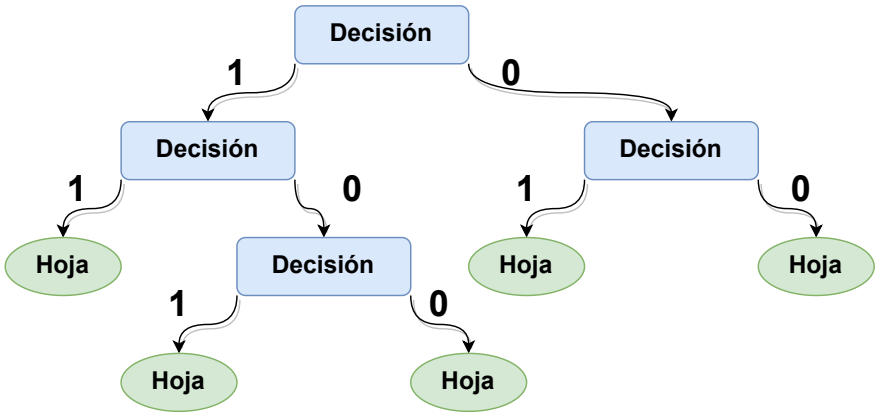


Figura 1-14: Esquema de árbol de decisión

Este tipo de modelos resaltan por la facilidad con que se pueden entender, además de su capacidad para trabajar tanto con variables cualitativas como cuantitativas. Estos modelos son proclives a un fenómeno llamado sobreajuste u *overfitting* que consiste en la incapacidad del modelo para predecir adecuadamente la respuesta a datos diferentes a los usados durante el entrenamiento, esto sucede debido a la sobresimplificación que pueden suponer algunos criterios de decisión, .

En el desarrollo del capítulo 7 se abordan la estructura y funcionamiento de los modelos de *Machine Learning* que resultan de especial interés para el desarrollo del presente trabajo.

1.7.2.3. Entrenamiento del modelo

Se conoce como entrenamiento en *Machine Learning* al proceso de alimentar un modelo con un gran volumen de datos, caracterizados mediante sus correspondientes *features* y *labels*, para que éste ajuste sus parámetros internos y pueda llevar a cabo la tarea para la que se entrena [36]. A medida que un modelo recibe datos de entrenamiento va ajustando sus parámetros internos de manera progresiva, por lo que es se aconseja que el conjunto de datos de entrenamiento sea tan numeroso como sea posible.

El entrenamiento exitoso de un modelo depende de un meticuloso proceso de diseño y selección del *dataset* o conjunto de datos. Se aconseja optar por *features* que guarden relación con la salida que se espera en el modelo. Se pueden emplear diversas técnicas de *EDA* para identificar y, de ser necesario, transformar los *features* que presenten las relaciones más estrechas con las salidas esperadas. El *dataset* debe dividirse en dos conjuntos: entrenamiento y validación, considerando que para el entrenamiento del modelo se deben usar tantas observaciones como sea posible. Se aconseja que el tamaño del *dataset* de entrenamiento sea considerablemente mayor que el de validación, algunas proporciones utilizadas para porcentajes de datos de entrenamiento y validación son 70/30 y 80/20. [37]

Además de la proporción, se debe cuidar el balance en los datos de entrenamiento y validación. Un *dataset* balanceado es aquel que tiene la misma cantidad de registros para cada una de las categorías en que podrá clasificar los datos el modelo. Un *dataset* desbalanceado puede sesgar el modelo y hacer que este clasifique erróneamente los datos que se le suministren. [38]

Cuando se dispone de un *dataset* suficientemente grande, el entrenamiento se puede realizar mediante la técnica de *hold-out* en la cual se separan los datos de validación se separan de los de entrenamiento, de forma que la validación se realice con datos que sea totalmente nuevos para el modelo.

En caso que no se cuente con un *dataset* numeroso para el entrenamiento de modelos, existen alternativas para el entrenamiento con *datasets* pequeños. Una de las más comunes es el entrenamiento de modelos usando la metodología *k-folds* [39] donde se utiliza repetidas veces el mismo *dataset* para realizar el entrenamiento y validación, cambiando en cada *fold* las muestras utilizadas para entrenamiento y validación como lo muestra la figura **1-15**. Los resultados de cada proceso individual de entrenamiento y validación (*fold*) se pueden sumar y promediar para obtener un equivalente a un único proceso de entrenamiento, o se pueden estudiar individualmente, según criterio de quien realice el proceso.

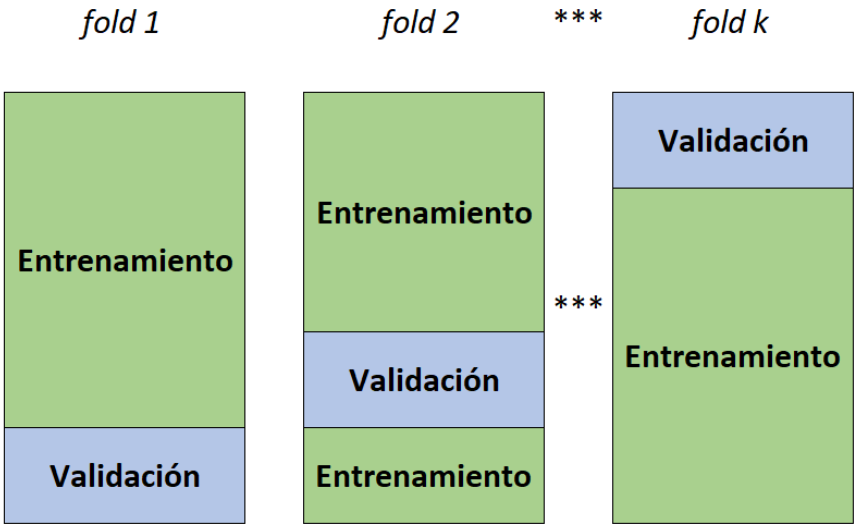


Figura 1-15: Esquema de división de datos en entrenamiento *k-folds*

Cuando el problema radica en el balance del dataset, se pueden emplear diferentes estrategias. A continuación se presentan algunas:

Tratamiento de datasets desbalanceados: Existen casos en los que se pueda ser inevitable tener *datasets* desbalanceados, este escenario es común, por ejemplo, en problemas de detección de fraude en transacciones bancarias como los presentados por Alarfa *et. al* [40] y Suhas *et. al* [41], donde la proporción de observaciones con el fenómeno de interés fue menor al 1 % en ambos casos.

Tres soluciones que se pueden implementar para manejar un dataset desbalanceado se mencionan a continuación:

- **Submuestreo:** Realizar un muestreo aleatorio de los datos pertenecientes a la categoría con mayor número de ocurrencias, de forma que se equilibren los *dataset* de entrenamiento y validación. Esta solución induce riesgos asociados a la pérdida de información del conjunto submuestreado y al uso de *datasets* con pocos elementos.

- **Sobremuestreo:** Consiste en añadir copias de la clase minoritaria al *dataset*. Esta aproximación puede inducir a errores de clasificación de la clase minoritaria debido a que el modelo puede memorizar las características de las observaciones repetidas en lugar de ajustarse para detectar similitudes.

- **Muestras sintéticas:** Utilización de algoritmos de interpolación como el SMOTE [42] para generar muestras sintéticas de la clase minoritaria. La efectividad de esta estrategia depende de la cantidad y calidad de las muestras disponibles en ambas clases.

Supóngase, por ejemplo, que se quiere entrenar un modelo para determinar si una imagen corresponde a un triángulo o a un cuadrado. Dependiendo del modelo y el *dataset* de entrenamiento, se pueden producir tres posibles escenarios en la capacidad del modelo para generalizar las características de los datos de entrenamiento, dicha capacidad se conoce como el ajuste del modelo y el proceso mediante el cual se produce el ajuste se conoce como aprendizaje. Los tres posibles tipos de ajuste que se pueden presentar se enumeran a continuación:

1. **subajuste o *underfitting*:** Sucede cuando el modelo entrenado no es capaz de generalizar las características de la tarea de clasificación o regresión según el caso, impidiendo que el modelo procese acertadamente nuevas entradas. El subajuste de un modelo puede tener diversas causas, entre las que se pueden encontrar: [43]

   - Se emplearon muy pocas muestras para entrenar el modelo.
   - Los *features* seleccionados no contienen información suficiente para que el modelo aprenda a realizar su tarea.
   - Las clases a la salida del modelo no presentan suficientes diferencias entre si, por lo que el modelo no es capaz de determinar claramente la pertenencia de una muestra a una u otra clase.

   En la figura **1-16** se muestra un caso de *underfitting* de modelo ocasionado porque las muestras del *dataset* de entrenamiento no contienen suficientes observaciones de los tipos de entradas que corresponden a triángulos.

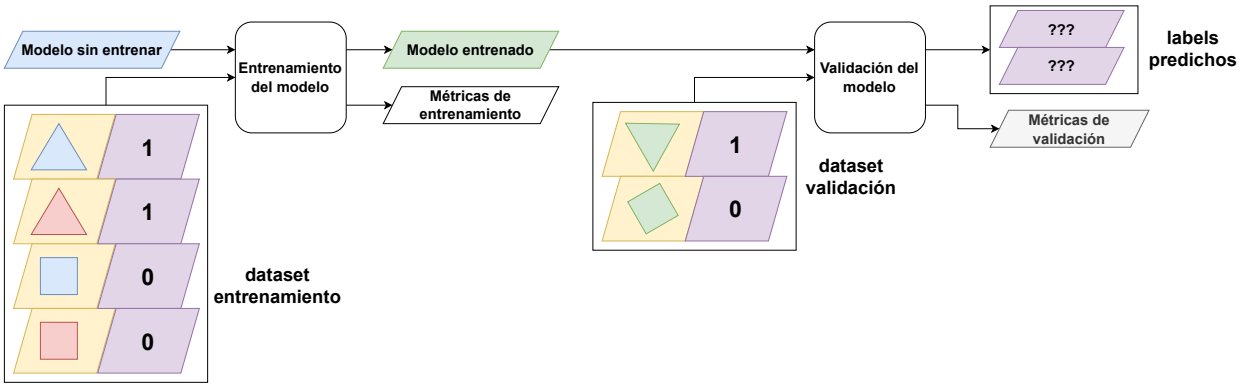


Figura 1-16: Esquema de ejemplo de caso de *underfitting*

2. **sobreajuste u *overfitting*:** El polo opuesto al *underfitting* sucede cuando el modelo se ajusta demasiado a ciertos tipos de asociación *feature-label*. En estos casos se dice que el modelo memoriza en lugar de aprender debido a que solamente predice la salida esperada ante datos de las mismas características que los datos usados en el entrenamiento, como en el ejemplo de la figura **1-17** donde el modelo se entrena para reconocer como triángulos solamente aquellos de color rojo y azul; al mostrarle al modelo un triángulo verde en la etapa de validación, este no es capaz de reconocerlo como triángulo y predice un *label* de 0 en lugar de un 1.

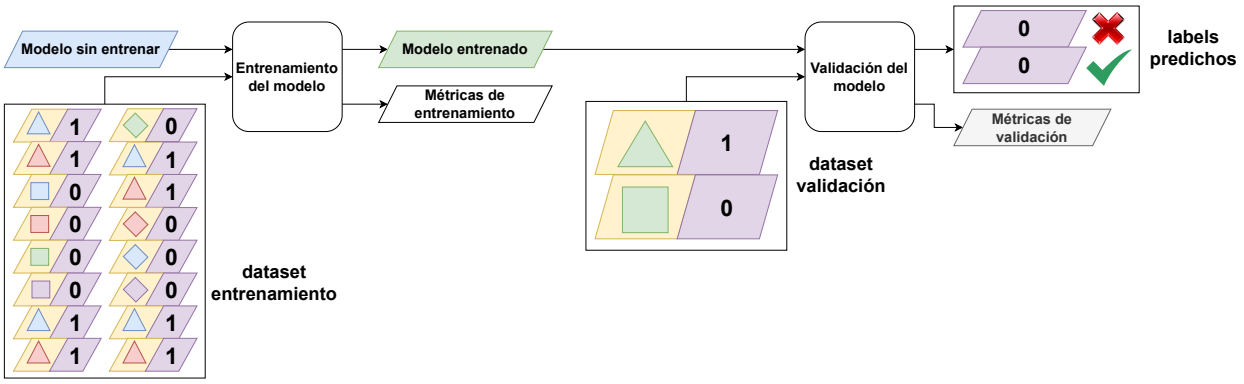


Figura 1-17: Esquema de ejemplo de caso de *overfitting*

3. **ajuste o *fitting*:** Este es el caso deseado posterior al entrenamiento de un modelo de *Machine Learning*. Esto sucede cuando el dataset de entrenamiento tiene la cantidad, variedad y calidad suficiente en los *features* como para que el modelo aprenda a identificar los patrones en los datos de entrada en lugar de memorizar. Es decir que el modelo entrenado va a ser capaz de reconocer e identificar correctamente los patrones de interés, incluso en datos que el modelo no haya visto previamente, como en el caso mostrado en la figura **1-18**.

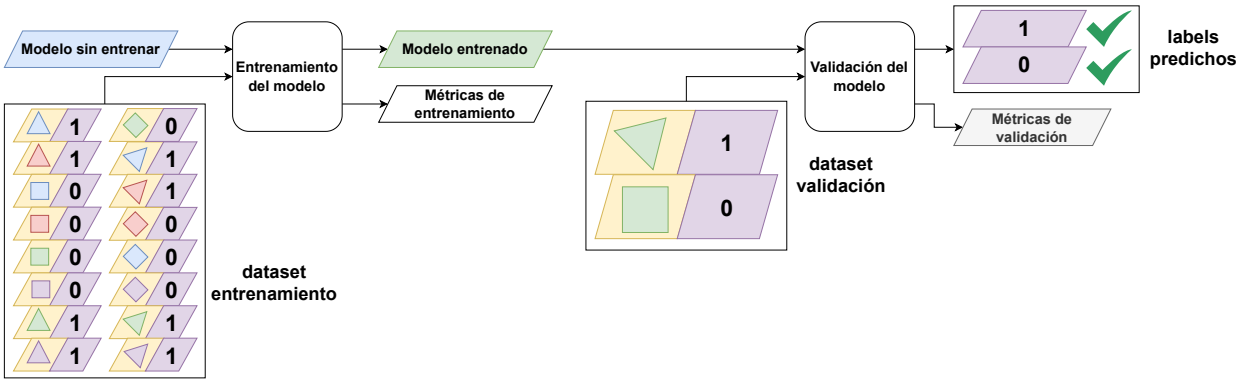


Figura 1-18: Esquema de ejemplo de caso de *fitting*

Se observa que en este caso el modelo es entrenado con variedad de colores y rotaciones tanto de triángulos como de cuadrados. Al llegar a la etapa de validación y mostrarle al modelo dos figuras con algunas características que no vistas en el entrenamiento, el modelo que ha aprendido correctamente a diferenciar triángulos de cuadrados y es capaz de etiquetar acertadamente cada figura.

1.7.2.4. Validación del modelo

Una vez se entrena el modelo, se realiza un proceso de validación para asegurarse de que el modelo entrenado funciona para el propósito deseado. Se le entregan al modelo entrenado los *features* del conjunto de validación, se registran las salidas producidas por el modelo y se comparan con los *labels* de validación. De la comparación entre el valor predicho por el modelo y el valor verdadero (*label* de validación) se tienen 4 posibles resultados, como se muestra en la tabla **1-1** para el caso de un *label* binario.

Nombre	Abreviatura	Predicción	Valor real
Verdadero Positivo	VP	1	1
Falso Negativo	FN	0	1
Verdadero Negativo	VN	0	0
Falso Positivo	FP	1	0

Tabla 1-1: Resultados de matriz de confusión de dos categorías

A partir del conteo de los casos mostrados en la tabla **1-1** se construye la matriz de confusión del modelo, como la mostrada en la figura **1-19**, que presenta gráficamente el resultado de la validación del modelo.

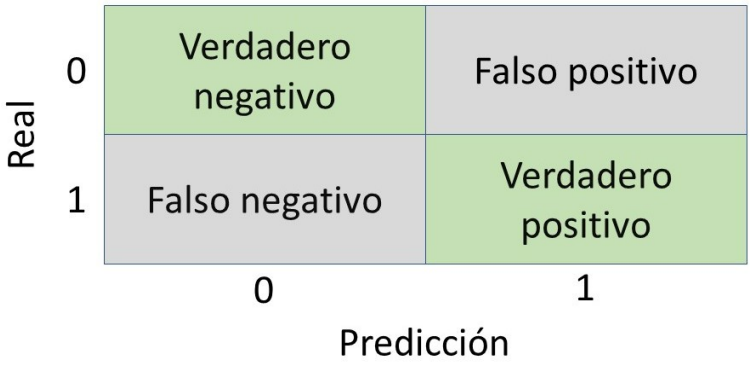


Figura 1-19: Estructura de matriz de confusión para *label* de 2 categorías

A partir de los conteos de la matriz de confusión se construyen métricas de desempeño del modelo, en las ecuaciones (5) a (7) se definen tres de ellas.

$$Sensibilidad = \frac{\sum VP}{\sum VP + \sum FN} \quad (5)$$

$$Especificidad = \frac{\sum VN}{\sum VN + \sum FP} \quad (6)$$

$$Precisión = \frac{\sum VP + \sum VN}{\sum VP + \sum FN + \sum VN + \sum FP} \quad (7)$$

2 Identificación del problema

Con base en los conceptos y la estructura expuesta en el capítulo 1, se identifica una oportunidad de interés en la manera en que se identifican fallos en los termopares del horno línea 1, con lo que se acota la definición del problema.

2.1. Identificación del problema

El monitoreo del estado de la red de termopares alrededor del horno representa un elevado coste logístico y operacional. La implementación de un modelo de *FDI* para los termopares del horno tiene el potencial de reducir la cantidad de tiempo que los operadores destinan al monitoreo del estado de salud de los sensores, permitiéndoles enfocarse en los casos en que el modelo identifique comportamientos anómalos en las señales.

Sin embargo, actualmente no se dispone de una línea base de conocimiento respecto a estados de normalidad o falla en los termopares del horno, por lo que se hace necesario establecer dicha línea base de forma que permita medir el desempeño de modelos de *FDI* enfocados en detectar la existencia de fallos en la red de termopares. El establecimiento de una línea base confiable de conocimiento se presenta como un hito para el desarrollo, evaluación, comparación y selección de modelos de *FDI* para los termopares del horno, por lo que se hace necesario resolver este desafío antes de proponer un modelo de *FDI*.

3 Objetivos

3.1. Objetivo General

Desarrollar un modelo basado en datos que permita identificar y aislar fallas en una red de termopares mediante técnicas basadas en datos.

3.2. Objetivos Específicos

1. Caracterizar y preprocesar el conjunto de datos disponible para seleccionar variables relevantes en el proceso de *FDI* de la red de termopares.
2. Generar métricas que permitan generar una línea base de conocimiento para relacionar variables del conjunto de datos con estados de falla de sensores en la red de termopares.
3. Diseñar una metodología que permita detectar y aislar fallas en la red de termopares utilizando técnicas basadas en datos.
4. Entrenar modelos basados en datos para detectar y aislar fallos en la red de termopares y evaluar su precisión.

4 Caracterización y preprocesamiento del conjunto de datos

4.1. Metodología

La primera fase de caracterización y preprocesamiento de los datos se realiza siguiendo el proceso desarrollado por Camacho et. al [44].

El sistema de monitoreo registra periódicamente diferentes variables de la planta con una periodicidad de 15 minutos. Se registran 1,180 variables o TAGs etiquetadas cronológicamente desde el 30 de septiembre de 2015 hasta el 30 de septiembre de 2019, conformando un total de 175,297 marcadores cronológicos y 206'850,460 registros individuales. Las variables monitoreadas, algunas obtenidas directamente de sensores y otras mediante cálculos, incluyen categorías como:

- Distancia entre pares de elementos.
- Flujo de calor.
- Flujo de agua.
- Intensidad de corriente.
- Masa transferida entre dos puntos.
- Modo de operación de sistemas.
- Potencia.
- Presión.
- Temperatura.
- Tensión.
- Velocidad.

De las 1,180 variables del sistema solamente una de ellas es de tipo categórico. Esta variable muestra instantes de tiempo en los que hubo fallas de comunicación y, adicionalmente, muestra el modo de control de la potencia eléctrica en los electrodos, el cual puede estar en modo manual o automático. Esta variable es codificada mediante *one-hot encoding* en una variable binaria con valor 0 para el modo manual, valor 1 para el modo automático o modo SAAE y *null* para los instantes de tiempo en que se registró falla de comunicación.

El proceso de limpieza de datos desarrollado se compone de 7 etapas, listadas a continuación y representadas en la figura **4-1**. Las variables que cumplen 1 o más condiciones de las listadas a continuación se eliminan del dataset.

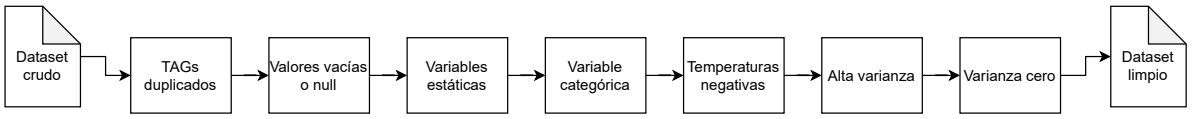


Figura 4-1: Diagrama de flujo de proceso de limpieza de datos

1. **TAGs duplicados:** Variables duplicadas o con información redundante. Esto se realiza comparando las variables entre si e identificando aquellas cuyos valores son exactamente iguales en todos sus registros. Se conserva una única variable de las identificadas como duplicadas.

2. **Valores vacíos o null:** Variables cuyo porcentaje de valores vacíos sea superior al 0.98 respecto al total de las observaciones disponibles.

3. **Variables estáticas:** Variables que almacenan un único valor y este no cambia en ningún instante de tiempo.

4. **Variable categórica:** La traducción de la variable categórica da lugar a instantes de tiempo en que se sabe que las medidas almacenadas no son ciertas, dado que se conoce la existencia de errores de comunicación. Los registros marcados como no confiables por fallas de comunicación son considerados como valores *null* y se aplica nuevamente el criterio de valores vacíos.

5. **Temperaturas negativas:** Valores de temperatura negativos no corresponden con el funcionamiento del termopar ni con estados de falla en el sensor, sino que pueden deberse a fallas de comunicación e interpretación de la información proveniente de los sistemas de adquisición de datos.

6. **Alta varianza:** Se asume, según información de manuales de operación los valores normales, valores de alarma y valores extremos de las señales. Los valores por fuera de estos rangos, más una tolerancia del 10 % respecto a los valores extremos son eliminados del conjunto de datos.

7. **Varianza cero:** Variables con un porcentaje de datos con varianza cero son eliminadas del dataset.

Posterior a la eliminación de las variables identificadas como candidatas por su baja calidad de información, se cambia el formato de las variables almacenadas de número en punto flotante de 64 bits a número de punto flotante de 32 bits por limitaciones del sistema de captura de datos.

4.2. Resultados y discusión

El proceso de limpieza de datos permite la identificación de variables que cumplen con los criterios de baja calidad de datos listados. La tabla **4-1** muestra la progresión de la reducción del tamaño del dataset a medida que se identifican variables con baja calidad de información.

Etapa	**Proceso**	**TAGs Identificados**	**TAGs después de filtro**
0	Dataset crudo	-	1,180
1	TAGs Duplicados	76	1,104
2	Valores vacíos o null	2	1,102
3	Variables estáticas	60	1,042
4	Variable categórica	5	1,037
5	Temperaturas negativas	74	963
6	Alta varianza	97	866
7	Varianza cero	22	844

Tabla 4-1: Proceso de eliminación de TAGs en dataset original con base en los criterios de calidad definidos.

Las variables se clasificaron de acuerdo con su participación en el proceso. Se crearon tres categorías principales:

- **Entrada:**
  1. Flujo de agua hacia los *Plate Coolers* y *Waffle Coolers*.
  2. Temperatura de agua después de enfriar *Plate Coolers* y *Waffle Coolers*.
  3. Altura de la pila de calcina.
  4. Composición química de la calcina.
  5. Flujo de calcina al horno.
  6. Temperatura de la calcina.
  7. Temperatura de líquidos de la calcina.

- **Operación:**
  1. Carga de calcina en el centro del horno.
  2. Carga de calcina en el semicentro del horno.
  3. Carga de calcina en la parte lateral del horno.
  4. Tensión de los electrodos.
  5. Corriente de los electrodos.
  6. Potencia de los electrodos.
  7. Impedancia de los electrodos.
  8. Posición de los electrodos.

- **Salida:**
  1. Composición química del metal.
  2. Composición química de la escoria.
  3. Expasión radial del horno.
  4. Expansión vertical del horno.
  5. Temperatura de la escoria.
  6. Temperatura de metal.
  7. Temperatura encima del centro del horno.
  8. Temperaturas de *Waffle Coolers*.

9. Temperaturas de *Plate Coolers*.
10. Temperatura de pared superior.
11. Flujos de calor en los *Plate Coolers* y *Waffle Coolers*.

El proceso de filtrado de datos permite la identificación y eliminación de un total de 340 TAGs. De los restantes 844 TAGs se conservan únicamente los TAGs de temperatura de *Waffle Coolers* y *Plate Coolers* con el fin de estudiar y comprender en detalle las relaciones que se puedan identificar entre dichas variables y que permitan establecer un modelo para identificar fallos en los sensores que miden dichas temperaturas.

Se tiene en total 322 TAGs correspondientes a temperaturas de *Waffle Coolers* y *Plate Coolers*, repartidos como se muestra en la tabla **4-2**.

Tipo de panel	**TAGs**
Plate Cooler A	37
Plate Cooler B	34
Plate Cooler C	36
Plate Cooler D	72
Waffle Cooler	143
Total	**322**

Tabla 4-2: Conteo y dristribución de TAGs de temperatura de *Waffle Coolers* y *Plate Coolers*

5 Generación de métricas

La correlación es una medida estadística que indica si existe una relación lineal entre dos variables de interés. El resultado de la correlación es un valor numérico cuya magnitud se encuentra entre -1 y 1, donde una correlación con valor 1 indica una relación de linealidad directa entre las variables y una correlación de -1 indica que la relación de linealidad es inversa. Una correlación de 0 indica que no existe relación directa ni inversa entre las dos variables estudiadas. La correlación entre dos variables se calcula mediante la ecuación (8):

$$r = \frac{\sum[(x_i - \bar{x})(y_i - \bar{y})]}{\sqrt{\sum(x_i - \bar{x})^2 * \sum(y_i - \bar{y})^2}} \qquad (8)$$

Donde x_i e y_i representan cada uno de los valores de las variables x e y respectivamente, y $\bar{x}$ e $\bar{y}$ representan los valores medios de x e y respectivamente.

Una matriz de correlación es la representación de la medida de correlación entre dos o más variables de interés, donde tanto filas como columnas representan cada una de las variables estudiadas y en el cruce la columna [n] con la fila [m] se muestra la correlación entre dichas variables. Dos aspectos importantes de las matrices de correlación:

- La diagonal principal de la matriz de correlación, donde se cruzan la fila y columna [n] muestra la correlación de una variable consigo misma, la cual siempre da como resultado el valor 1.

- El valor de correlación de la fila [n] con la columna [m] es el mismo que el de la fila [m] con la columna [n], lo cual se puede deducir de la ecuación (4).

La figura **5-1** muestra la matriz de correlación (izquierda) y una representación de dicha matriz de correlación en un mapa de calor (derecha) para un conjunto de 5 *Waffle Coolers* consecutivos seleccionados aleatoriamente. Se empleó la totalidad de datos existentes en el tiempo para el cálculo de la matriz de correlación.

	Waffle Cooler 32 B Sur Oeste	Waffle Cooler 32 A Sur Oeste	Waffle Cooler 33 A Sur Oeste	Waffle Cooler 33 B Sur Oeste	Waffle Cooler 34 A Sur Oeste
Waffle Cooler 32 B Sur Oeste	1.000000	0.976466	0.054861	0.972328	0.916414
Waffle Cooler 32 A Sur Oeste	0.976466	1.000000	0.043883	0.980808	0.911205
Waffle Cooler 33 A Sur Oeste	0.054861	0.043883	1.000000	0.051994	0.041873
Waffle Cooler 33 B Sur Oeste	0.972328	0.980808	0.051994	1.000000	0.918700
Waffle Cooler 34 A Sur Oeste	0.916414	0.911205	0.041873	0.918700	1.000000

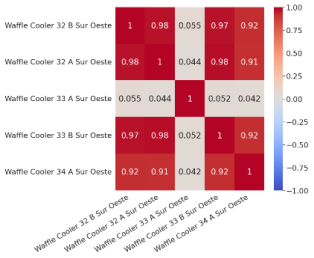


Figura 5-1: Ejemplo de matriz de correlación (izq) y representación en mapa de calor (der).

5.1. Uso de termopares cercanos como alternativa a redundancia física

Dado que, salvo algunos *Waffle Coolers*, no existe más de un termopar instalado en cada panel, se busca encontrar medidas que ofrezcan tanta similitud como sea posible a la redundancia física en los termopares del horno.

Para efectos del presente trabajo, se denominará **vecindario** a cualquier grupo de termopares que se estudian en conjunto.

5.1.1. Metodología

La metodología presentada en el presente aparte fue compartida inicialmente con Cerro Matoso S.A a través del documento: informe de Actividad 2.6: Informe de correlación y Redundancia [45]. En el informe se presentan avances preliminares de la metodología desarrollada y se presentan las potenciales aplicaciones de la metodología con la asociación a estados de falla en vecindarios de termopares.

Se calcula la matriz de correlación en vecindarios conformados según diferentes configuraciones y se estudia el valor de la correlación entre los miembros del grupo. Se estudian tres configuraciones de agrupación.

- Por posición angular
- Por tipo y nivel de panel.
- Todos los paneles adyacentes a un panel central.

Se estudia el comportamiento de la matriz de correlación para diferentes grupos de termopares cercanos. Se busca determinar el tipo de agrupación de termopares que ofrezca la correlación más alta entre todos los miembros del grupo, partiendo de la consideración de que la correlación entre una variable con sigo misma tiene valor 1, lo que se presentaría en una medida redundante de dos sensores.

Posteriormente se realiza un estudio de la capacidad de detección de cambios en la correlación entre las medidas de los termopares de un vecindario en ventanas de tiempo de diferente ancho. Se presenta una comparación de los resultados obtenidos al realizar un barrido de parámetros del ancho de ventana, con lo que se determinan anchos de ventana de interés para el proyecto.

5.1.2. Vecindarios de paneles agrupados por posición angular

Se inicia la exploración agrupando todos los *Plate Coolers* y *Waffle Coolers* en una misma posición angular. Como ejemplo se muestra el vecindario agrupado en la posición 12 (letra L) del cuadrante nor-oeste, este vecindario es tomado aleatoriamente y su ubicación en el horno se muestra gráficamente en la figura **1-3**, como se muestra en la figura **5-2**.

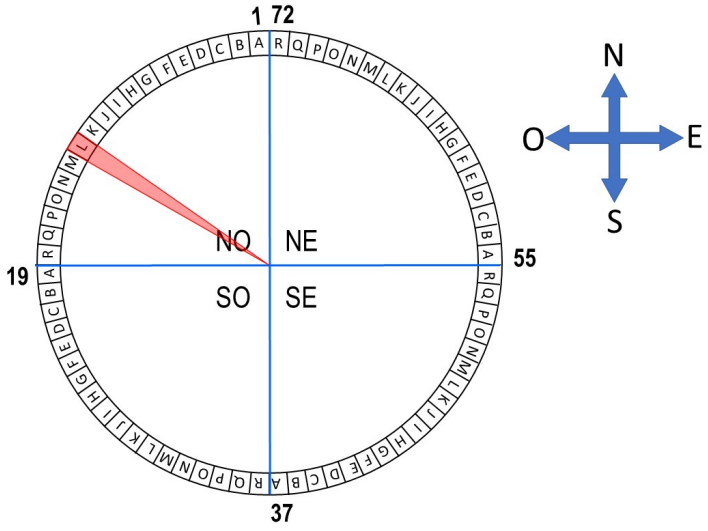


Figura 5-2: Ubicación de paneles en posición 12

La figura **5-3** se obtiene graficando un extracto de los valores registrados para los paneles existentes en la posición angular 12. Se aprecia, en primera medida, similitud entre las diferentes señales y su única diferencia notable es el valor promedio (valor DC) de las señales.

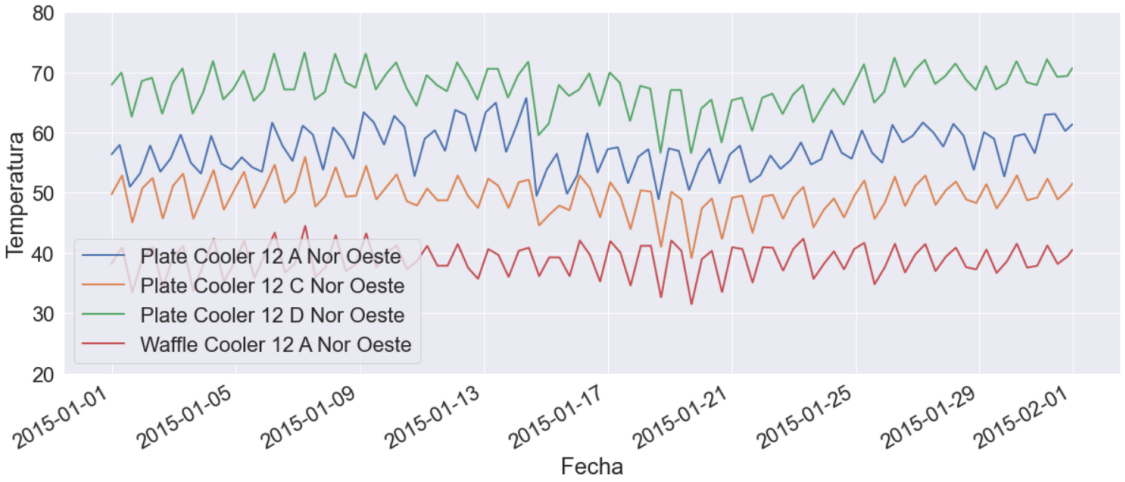


Figura 5-3: Extracto de línea de tiempo de paneles en la posición angular 12

Al calcular la matriz de correlación del vecindario bajo estudio, mostrada en la figura **5-4**, se encuentra que solo existe correlación alta (cercana a 1) para paneles de un mismo tipo a pesar de encontrarse situados en la misma o muy cercana posición angular. Este comportamiento se repite independiente de la posición angular que se tome como punto de partida para agrupar termopares cercanos en vertical.

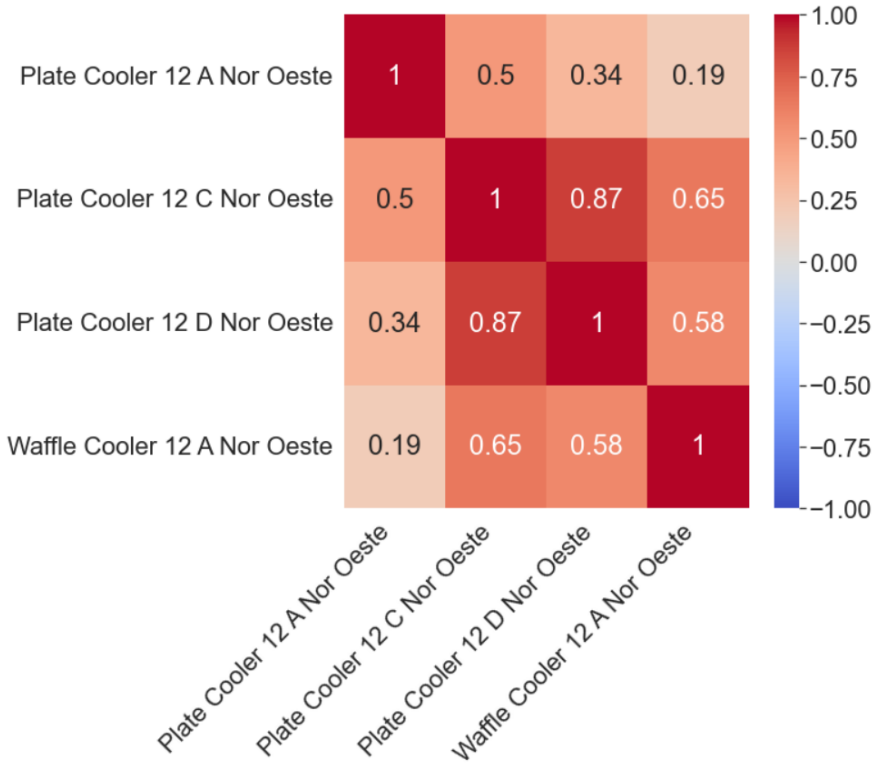


Figura 5-4: Matriz de correlación de paneles en la posición angular 12

Esta primera agrupación de vecindarios presenta, salvo la diagonal, solamente un valor cercano a 1, correspondiente a la correlación entre los *Plate Coolers* de los niveles C y D. Distribuciones y valores similares se identificaron para diversos vecindarios agrupados en torno a una misma posición angular en los 4 cuadrantes del horno.

Se continúa con el estudio de las configuraciones de agrupación restantes para obtener resultados de los tres casos propuestos.

5.1.3. Vecindarios de paneles agrupados por tipo y nivel de panel

En este segundo tipo de agrupación, se agrupan los termopares pertenecientes a un mismo tipo y nivel. Se inicia analizando vecindarios de *Waffle Coolers* y posteriormente se estudian vecindarios conformados por *Plate Coolers* agrupados un nivel a la vez.

Para evaluar esta posibilidad se inicia graficando las medidas de todos los termopares del horno en un instante de tiempo aleatorio, separándolos por niveles. Para facilitar la visualización, se grafican los datos sobre una geometría polar y se superponen sobre el esquema de ubicación de paneles en el horno mostrado en la figura **1-3**. Se le da el nombre de plot de radar. La figura **5-5** muestra un plot de radar de *Plate Coolers* y la figura **5-6** muestra un plot de radar de *Waffle Coolers*.

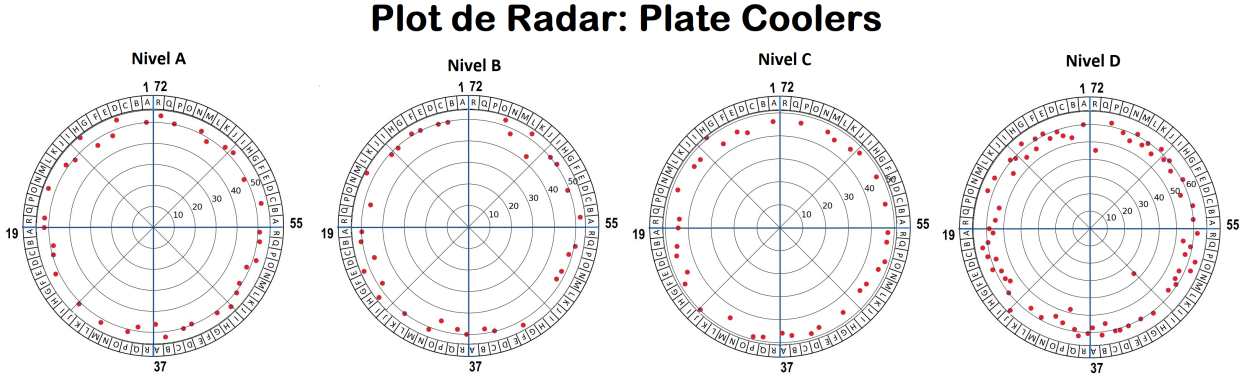


Figura 5-5: Plots de radar de *Plate Coolers*

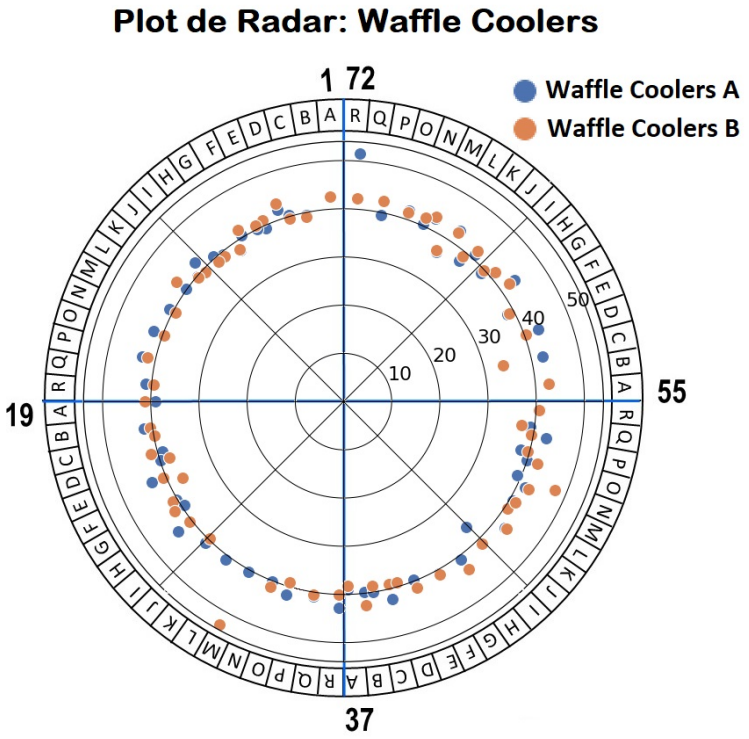


Figura 5-6: Plots de radar de *Waffle Coolers*

Se aprecia que las medidas de temperatura entre paneles cercanos de un mismo nivel presentan similar valor instantáneo, salvo algunos valores puntuales que podrían ser *outliers*[1]. Este comportamiento se observa tanto en *Plate Coolers* como en *Waffle Coolers* en diversos instantes de tiempo aleatorios empleados. En el caso particular de los *Waffle Coolers* se observa que los paneles que tienen dos termopares (A y B) presentan, generalmente, similitud en la medida obtenida de los dos sensores.

Con estos resultados preliminares se inicia el estudio de la conformación de vecindarios por tipo y nivel de panel.

5.1.3.1. Matrices de correlación de Waffle Coolers

Se calcula la matriz de correlación de todos los *Waffle Coolers* en la pared del horno y se visualiza en la figura **5-7**. Se identifica la existencia tanto de zonas de alta correlación (color rojo) como zonas de correlación cero y ligeramente menor a cero (colores blanco y azul tenue) en torno a la diagonal principal de la matriz.

[1]Observación inusual muy fuera del rango de valores esperado.

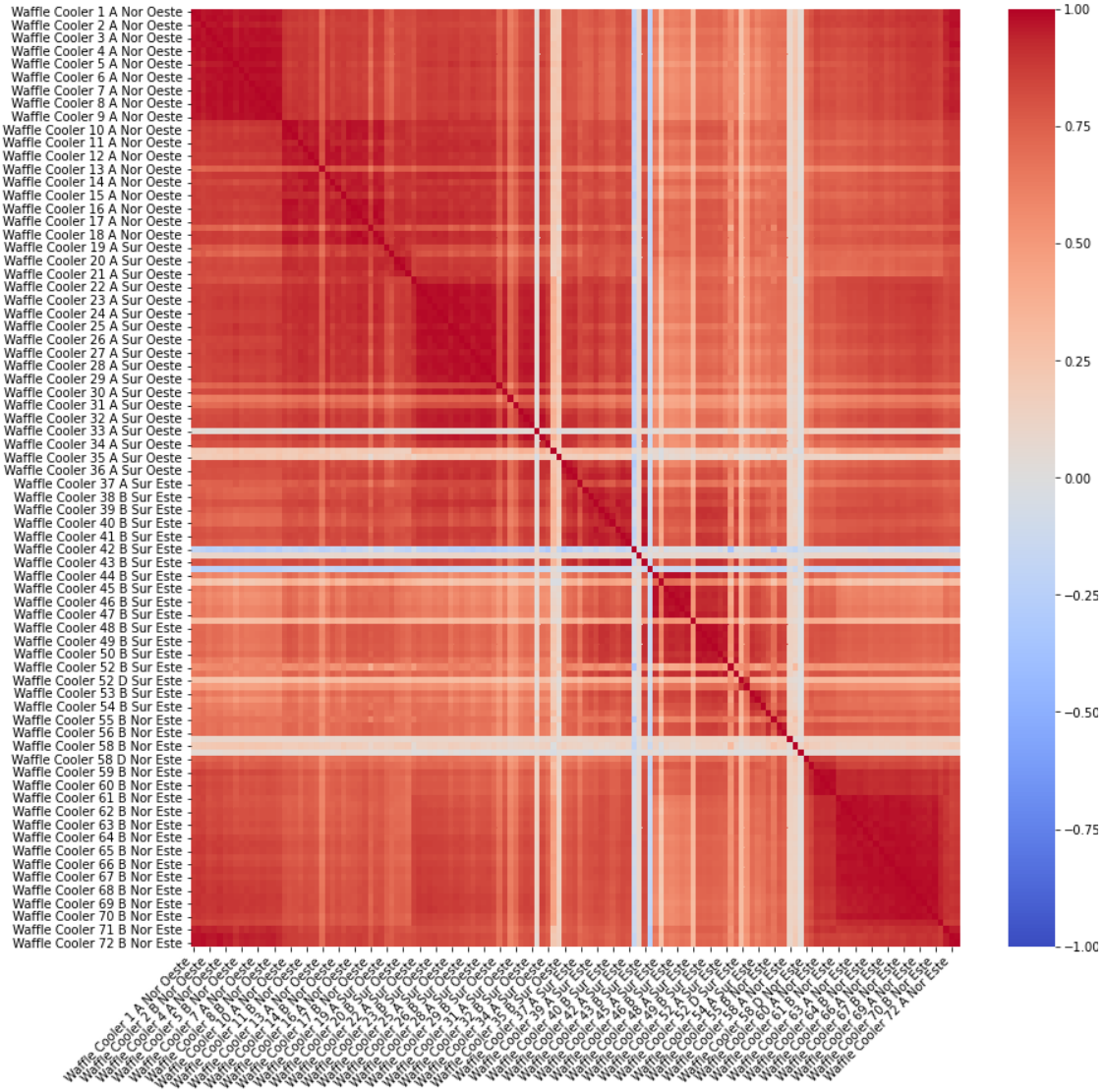


Figura 5-7: Matriz de correlación de *Waffle Coolers*

Las zonas de alta correlación alrededor de la diagonal principal sugieren la posibilidad de emular redundancia física entre termopares considerando grupos físicamente cercanos dentro de un mismo nivel. Se centra la atención en las zonas de baja correlación para estudiar el comportamiento de los datos en estas variables. Para ello se reduce el conjunto de datos a una región que contiene termopares de baja correlación, como se muestra en la figura **5-8**.

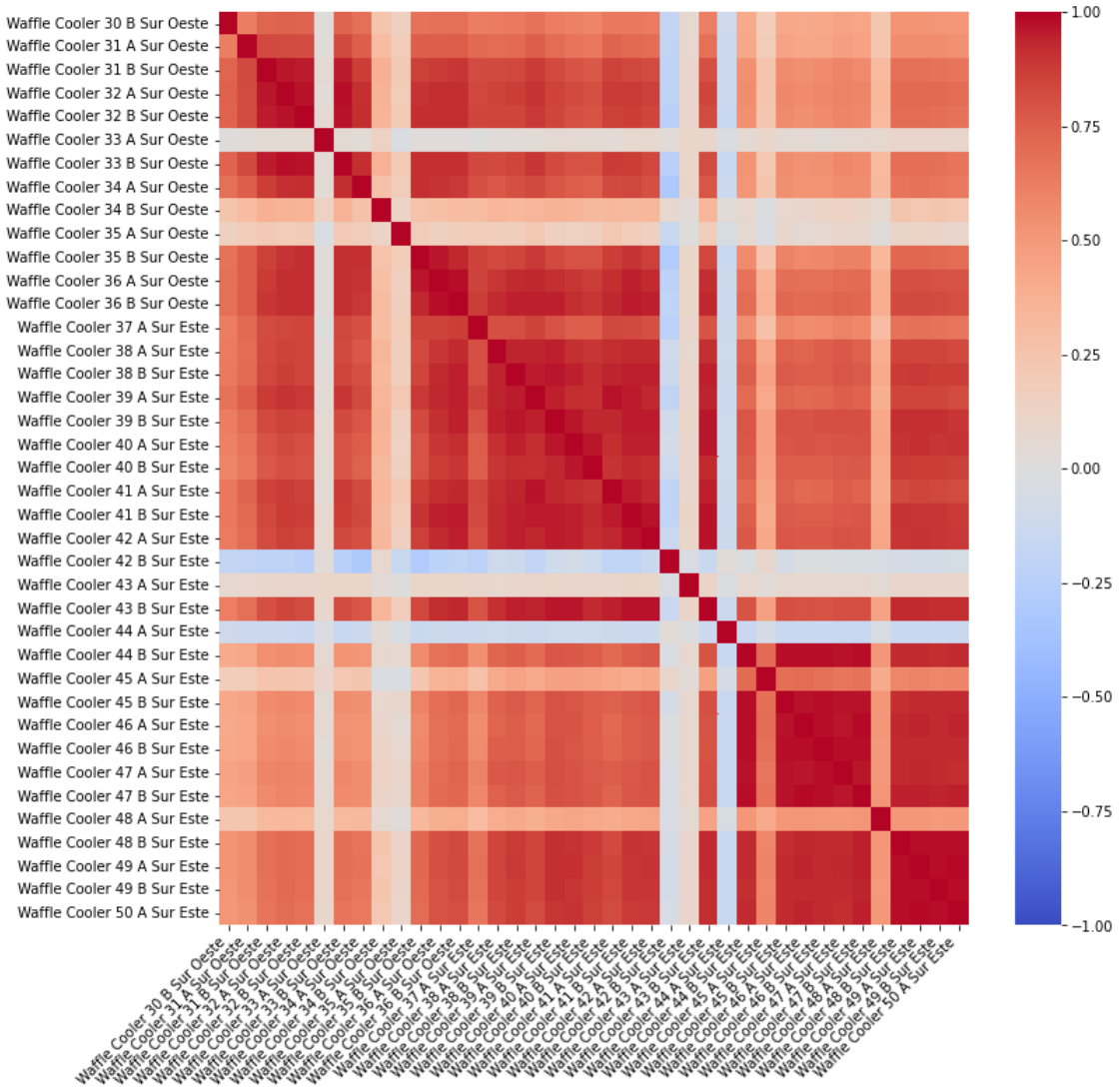


Figura 5-8: Acercamiento de matriz de correlación de *Waffle Coolers*

El vecindario de la figura **5-8** se compone por los *Waffle Coolers* entre las posiciones angulares 30 (Sur Oeste) y 50 (Sur Este), como se muestra en la figura **5-9**. Se selecciona el primer TAG o variable con baja correlación, que corresponde al *Waffle Cooler* 33 A, ubicado en la posición angular 33. Se seleccionan termopares físicamente cercanos a este y se calculan 8 variables estadísticas descriptivas de las variables, los cuales se muestran en la figura **5-10**.

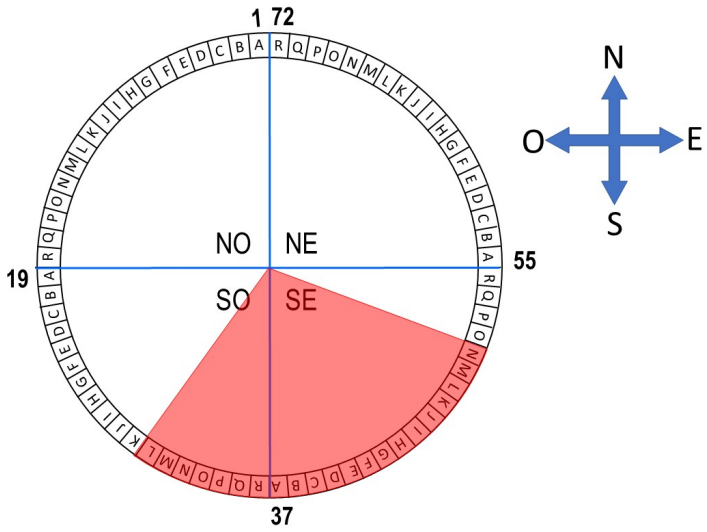


Figura 5-9: *Waffle Coolers* agrupados entre posiciones angulares 30 y 50

Se evidencia que el TAG de interés (412_TR12723OA) presenta valores mínimos y máximos por fuera de aquellos encontrados en su operación normal. Especialmente la presencia de valores negativos advierte que existe una anomalía con los datos almacenados para esta variable.

	Waffle Cooler 32 B Sur Oeste	Waffle Cooler 32 A Sur Oeste	Waffle Cooler 33 A Sur Oeste	Waffle Cooler 33 B Sur Oeste	Waffle Cooler 34 A Sur Oeste
conteo	175297.000000	175297.000000	175297.000000	175297.000000	175297.000000
media	43.772020	43.512099	69.003798	44.015902	43.337592
desv. estandar	3.675964	3.516178	173.515718	3.940244	3.734734
min	32.285545	31.757343	-270.000000	31.866142	-270.000000
perc .25	41.181961	41.059788	41.031914	41.265999	40.946838
perc .50	43.786896	43.592922	44.394524	44.065548	43.176418
perc. .75	46.268185	45.949207	48.084930	46.646374	45.412060
max	70.844902	72.798248	1372.000000	81.554024	67.966957

Figura 5-10: Análisis descriptivo de variables de vecindario alrededor de *Waffle Cooler* 33 A

La presencia de valores extremos en los mínimos y máximos del *Plate Cooler* 33 A sugiere la presencia de una anomalía. Al comparar los valores cuartiles de las variables, se evidencia que sus valores son muy similares entre sí, con una diferencia menor al 10 % en todos los casos. Al graficar los datos de las 5 variables en una línea de tiempo, como se aprecia en la figura **5-11**, se identifica que los valores extremos corresponden a un comportamiento desde inicios de 2018.

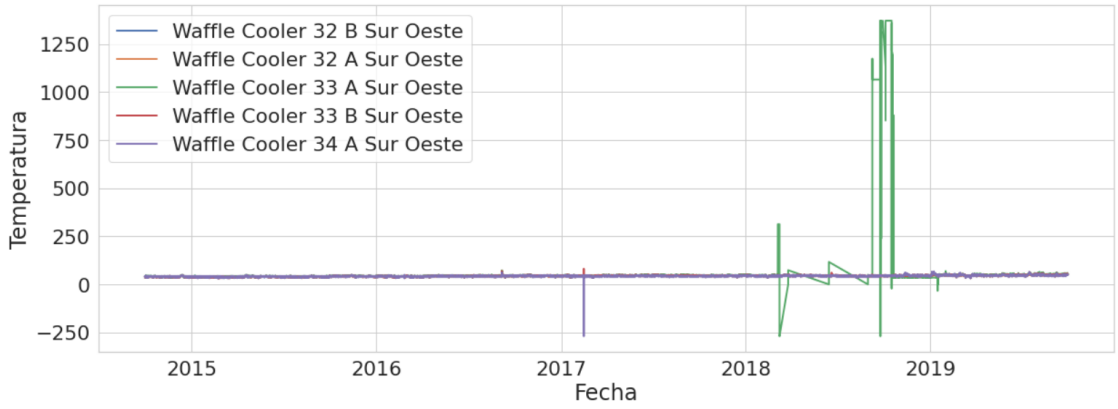


Figura 5-11: Línea de tiempo de variables alrededor de *Waffle Cooler* 33 A

Se calcula la matriz de correlación en dos escalas de tiempo diferente, una incluyendo los valores con fallos y otra recortando la línea de tiempo de forma que no se incluyan los valores con fallos. Los resultados se presentan en la figura **5-12**.

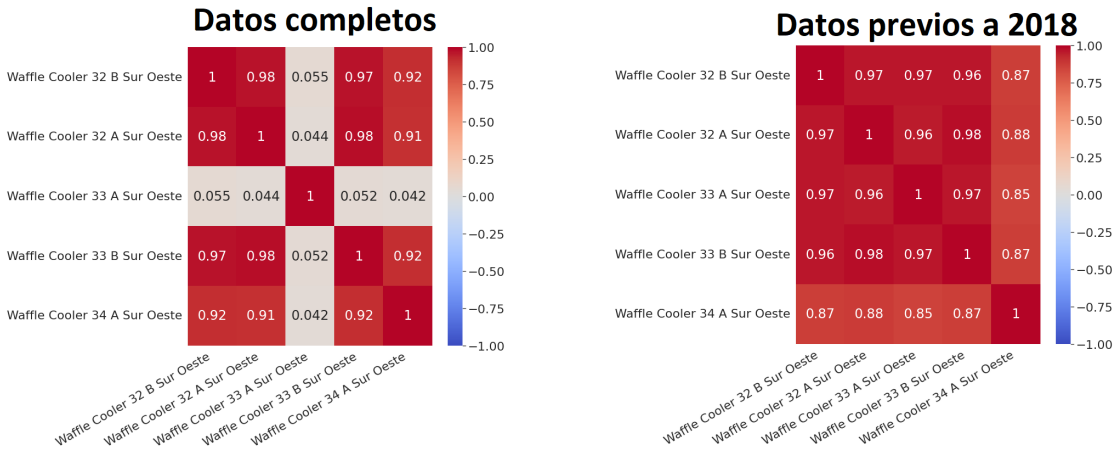


Figura 5-12: Comparación de matrices de correlación de vecindario en torno al *Waffle Cooler 33 A*

La correlación del *Waffle Cooler* 33 A mejora notablemente considerando solamente los datos previos a 2018, pasando de valores cercanos a 0 a valores cercanos a 1 en casi todos los casos.

Se observa en la figura **5-12** que la correlación del *Waffle Cooler* 34 A es, en comparación, apreciablemente menor que la de los demás miebros del vecindario que se está estudiando. Al remitirnos a la figura **5-11**, se observa que el *Waffle Cooler* 34 A tiene un pico negativo con valor de -250 °C antes del inicio del año 2017.

Se realiza un experimento calculando las matrices de correlación de los datos acotados hasta antes del año 2017 y se compara con los resultados obtenidos en la figura **5-12** al acotar los resultados hasta antes del año 2018. Los resultados se muestran en la figura **5-13**.

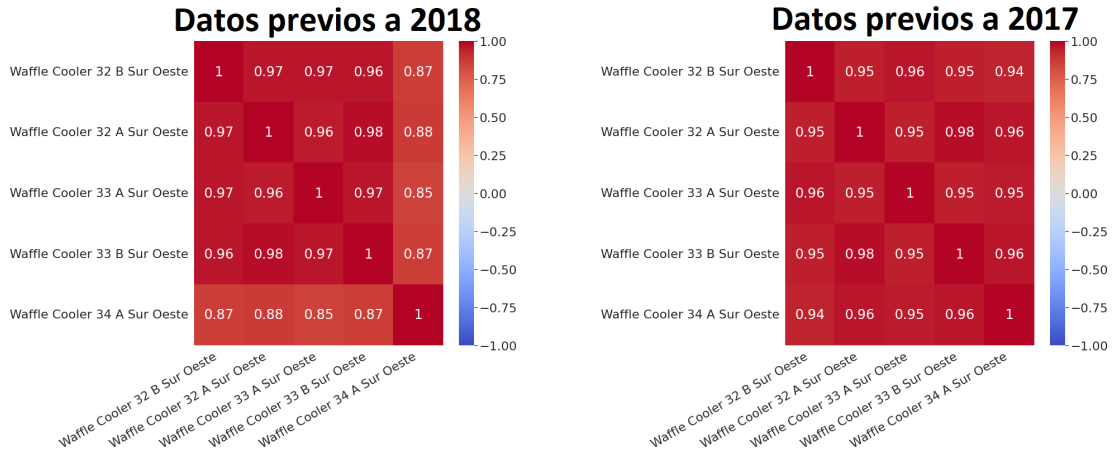


Figura 5-13: Comparación de matrices de correlación de vecindario en torno al *Waffle Cooler 33 A* con líneas de tiempo acotadas a 2017 y 2018

Se evidencia que la matriz de correlación de los datos previos al año 2017 presenta una correlación generalizada cercana a 1 en todos sus componentes. Comportamientos similares se identificaron en diferentes vecindarios y escalas de tiempo donde, salvo la existencia de fenómenos como outliers, variables congeladas en un valor o fenómenos transitorios, existe una conexión entre valores altos de correlación y similitud entre termopares agrupados en vecindarios por tipo y nivel de panel.

Considerando el resultado mostrado, el uso de la correlación en vecindarios de termopares agrupados por tipo y nivel puede ayudar a detectar la presencia de fallos en un sensor particular, dado que bajo condiciones normales de operación, las señales de dos *Waffle Coolers* cercanos guardan tal similitud que sus valores de correlación son cercanos a 1.

5.1.3.2. Matriz de Correlación de Plate Coolers

Los resultados obtenidos en 5.1.3.1 a sugieren que se podría esperar una medida alta de correlación entre vecindarios de *Plate Coolers* agrupados por nivel.

Se inicia calculando la matriz de correlación de todos los *Plate Coolers*. El resultado se muestra en la figura **5-14**.

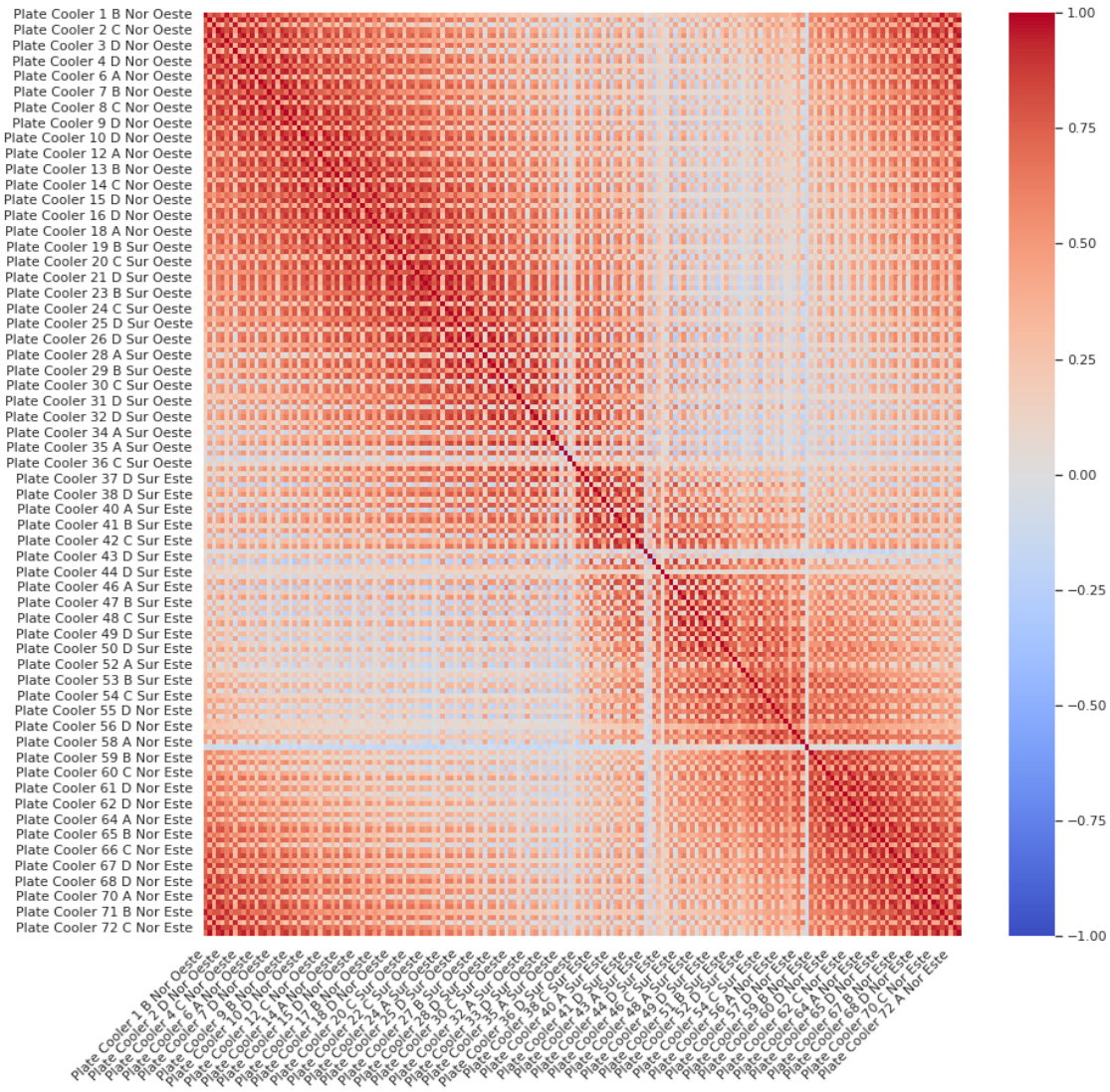


Figura 5-14: Matriz de correlación *Plate Coolers*

A diferencia de lo evidenciado en la figura **5-7** de los *Waffle Coolers*, y como se esperaba, no se evidencia alta correlación en torno a la diagonal principal. Por lo que se segmentan los TAGs por nivel y se calcula la matriz de correlación para cada nivel de *Plate Coolers*, como se muestra en la figura **5-15**.

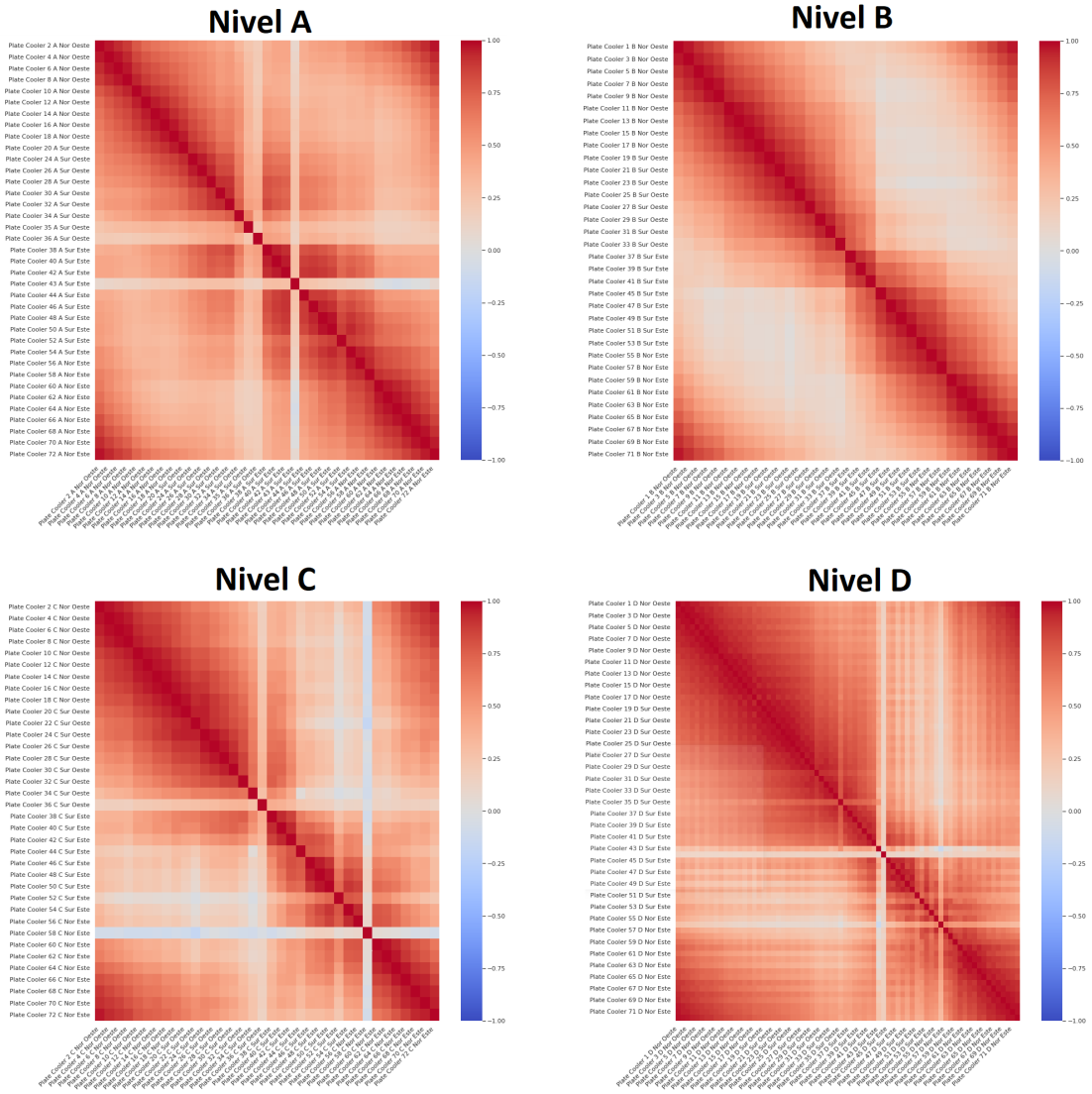


Figura 5-15: Matriz de correlación de *Plate Coolers* por nivel

En las 4 matrices se observa que existe alta correlación (color rojizo) en torno a la diagonal principal de cada matriz.

Se estudia vecindario alrededor del *Plate Cooler* 43 A. Al realizar un análisis similar al realizado en 5.1.3.1, se encuentra que la baja correlación se debe fenómenos como Valores extremos, Variables ancladas y Comportamientos transitorios en el comportamiento de un panel específico, lo cual coincide con los comportamientos observados en 5.1.3.1 para los *Waffle Coolers*.

Se calcula la matriz de correlación del vecindario con los valores sin filtrar, y luego se calcula una segunda matriz de correlación filtrando manualmente los valores extremos y los valores anclados. El resultado se muestra en la figura **5-16**.

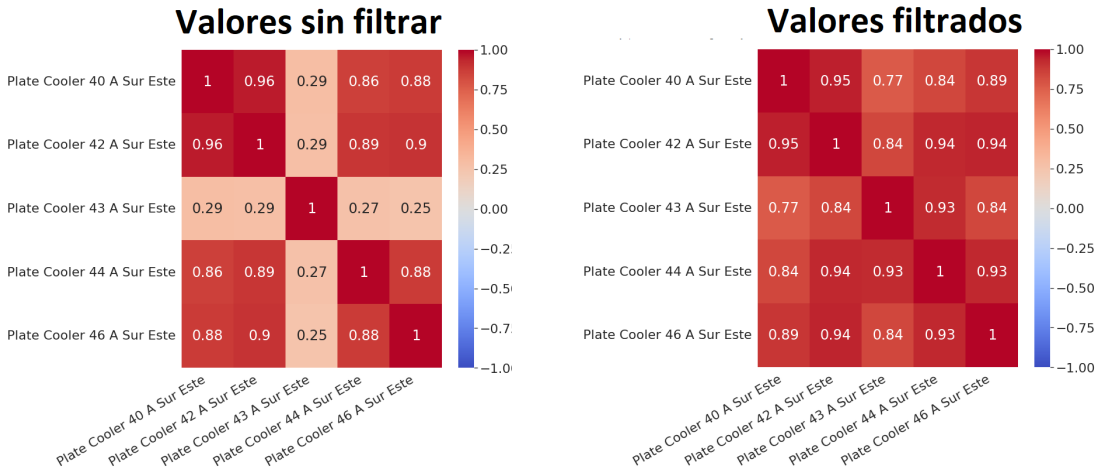


Figura 5-16: Comparación de matrices de correlación de *Plate Coolers* alrededor del panel 43 A con filtrado de datos

Se observa que, análogo a lo observado para los *Waffle Coolers*, a medida que se filtran anomalías en las señales, se evidencian valores de correlación altos para vecindarios de paneles agrupados por tipo y nivel.

5.1.4. Vecindarios de paneles agrupados por adyacencia a un panel central

Se seleccionan vecindarios agrupados en torno a un panel central, como se muestra en la figura **5-17**, donde se agrupan en color azul los termopares alrededor de un termopar central, resaltado en color verde. En cada caso, se usan tantos termopares como vecinos adyacentes se tengan disponibles.

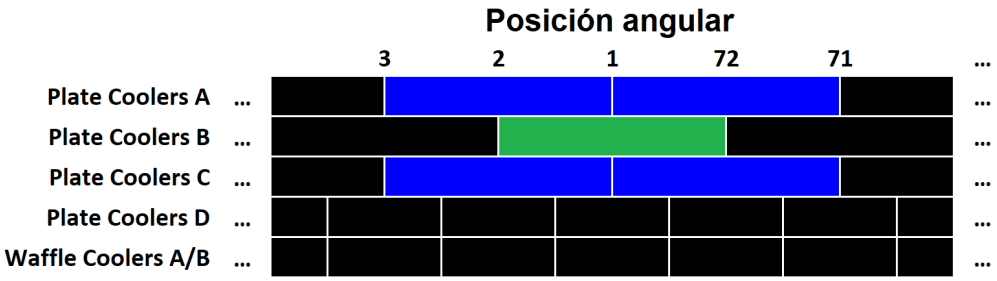


Figura 5-17: Ejemplo de vecindario de termopares adyacentes

Se estudia un vecindario con el *Plate Cooler* 65 B como panel central. El vecindario lo componen los *Plate Coolers*: 64 A, 66 A, 65 B (central), 64 C y 66 C. La forma en que se agrupa el vecindario estudiado es la misma que en la figura **5-17**, cambiando únicamente la posición angular de los paneles agrupados. La figura **5-18** ilustra la posición angular de los vecindarios agrupados para el estudio de vecindario formado por temopares adyacentes.

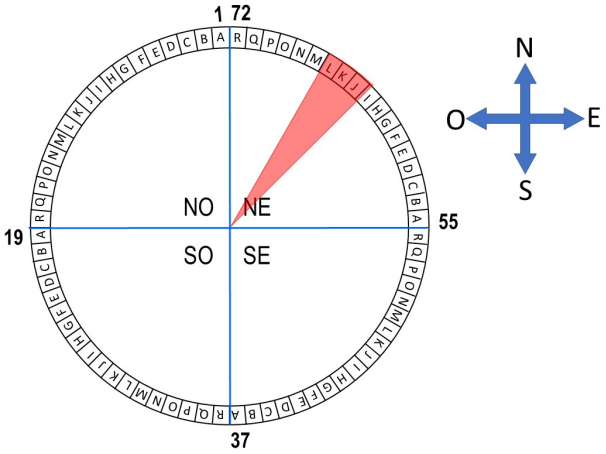


Figura 5-18: Posición angular de vecindario de termopares adyacentes

Se calcula la matriz de correlación de los termopares en el vecindario de adyacentes. El resultado, mostrado en la figura **5-19**, muestra que la correlación del panel central con los paneles del nivel C es considerablemente más baja que la correlación con los paneles del nivel A.

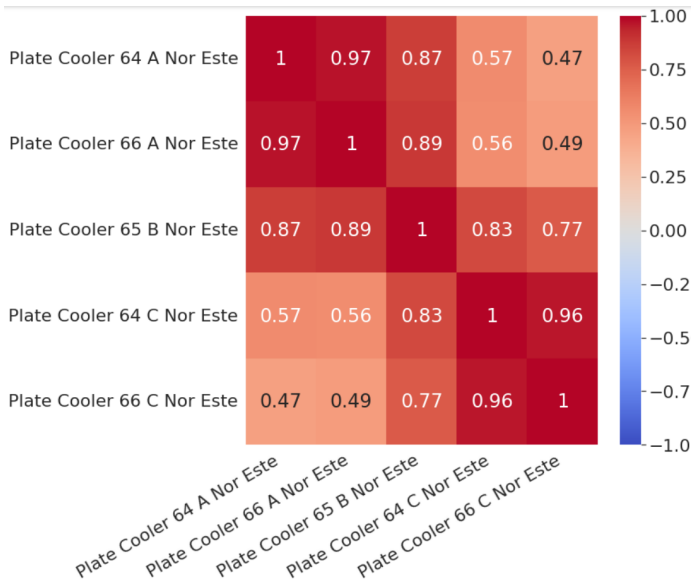


Figura 5-19: Matriz de correlación de vecindario de termopares adyacentes

Se realiza una prueba rápida acotando la línea de tiempo de los datos usados para calcular la matriz de correlación entre los miembros del vecindario. Se calcula nuevamente la matriz de corelación entre los miembros del vecindario considerando únicamente los datos entre el 2017-01-01 y el 2017-06-06. La comparación se muestra en la figura **5-20**.

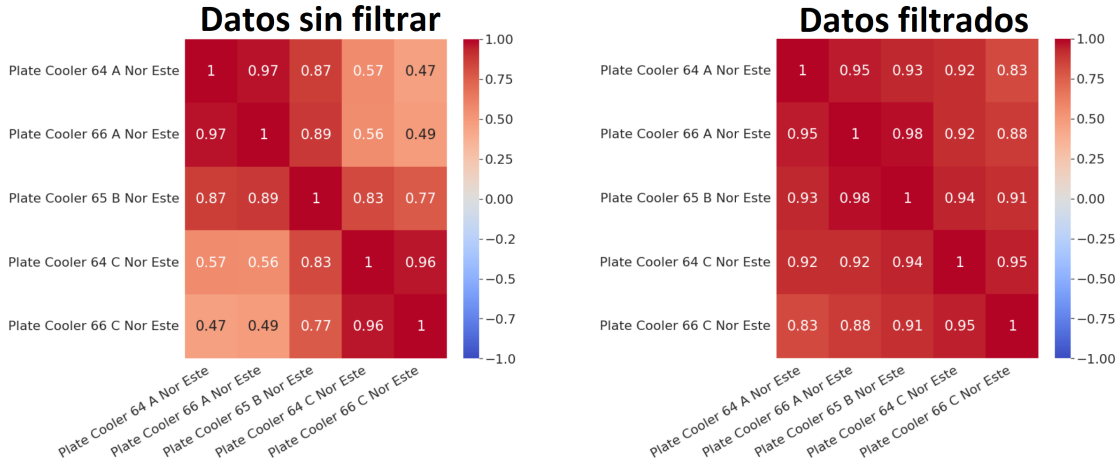


Figura 5-20: Matriz de correlación de vecindario de termopares adyacentes con datos filtrados

Se observa una notable mejoría en la correlación entre los diferentes miembros del vecindario, por lo que esta configuración para agrupar termopares se considera también como una candidata viable.

5.1.5. Resultados y discusión

Se tiene finalmente que las configuraciones de vecindarios por tipo y nivel y por adyacencia presentan resultados similares en cuanto a que en ambos casos se tienen valores de correlación cercanos a 1 entre todos los miembros del vecindario cuando las señales de los termopares no presentan fallos.

De las dos configuraciones posibles de vecindarios, se decide continuar con la agrupación de paneles por tipo y nivel. La razón fundamental de esta elección es que el progresivo cambio en las propiedades físicas, químicas y termodinámicas del material que reacciona al interior del horno, a medida que se deposita al fondo del mismo, puede generar diferencias en la temperatura a lo largo del eje vertical en una misma posición angular. La configuración de vecindarios de termopares de un mismo tipo y nivel no garantiza que las medidas sean idénticas para los termopares libres de fallos, pero reduce la probabilidad de la aparición de diferencias de temperatura debidas a la deposición de metal fundido.

5.2. Análisis en ventanas de tiempo

El análisis de matrices de correlación permite identificar anomalías en medidas registradas por vecindarios de sensores. Sin embargo, para que estos fallos sean apreciables en el cálculo de la matriz de correlación, deben ser suficientemente significativos dentro de la totalidad de los datos como para generar cambios sensibles en la varianza de la totalidad del conjunto de datos.

En algunos casos las variaciones en los datos son solamente perceptibles al estudiar intervalos cortos de tiempo, del orden de horas o días, como el mostrado en la figura **5-21** que sucede entre mediados del 2018-10-26 y mediados del 2018-10-29.

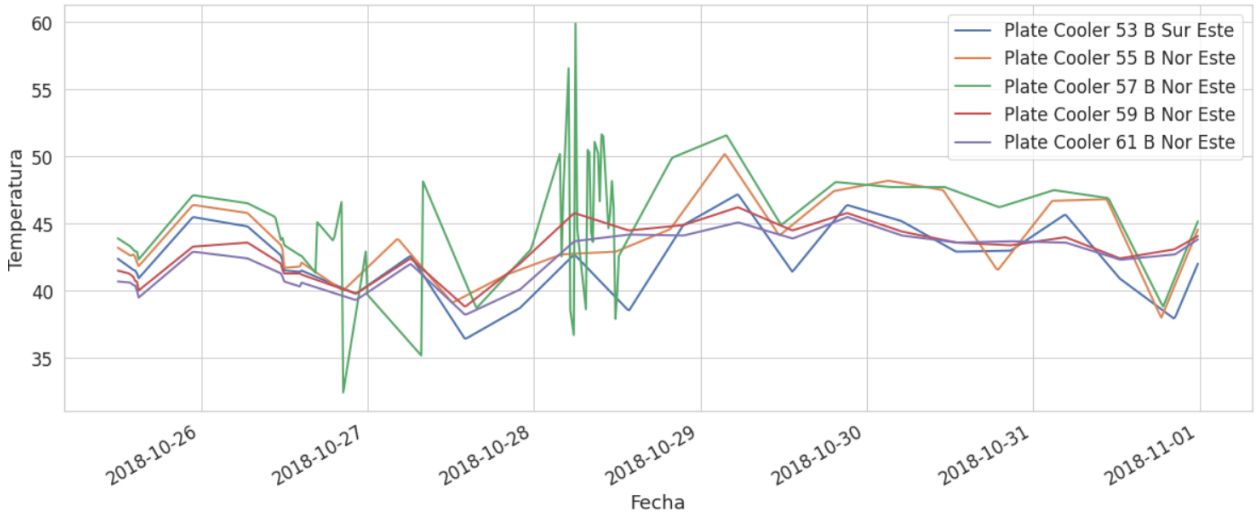


Figura 5-21: Ejemplo de fallo de corta duración

5.2.1. Metodología

Se selecciona la línea de tiempo de un vecindario seleccionado, se segmenta en ventanas de tiempo de ancho fijo como se muestra en la figura **5-22** y se estudia el cambio de la correlación y de la diferencia de temperatura entre las variables del vecindario en cada ventana.

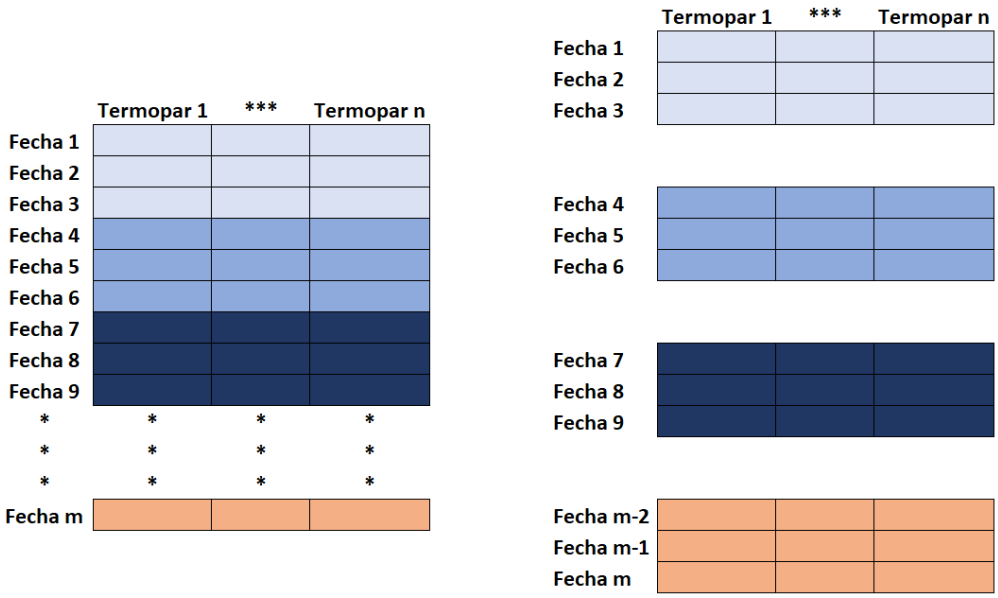


Figura 5-22: Ejemplo de división de un conjunto de datos en ventanas

De cada ventana de tiempo se obtiene una matriz de correlación. Para estudiar el cambio de la correlación entre las variables en el tiempo, se construyen líneas de tiempo del valor de la correlación entre cada par de variables del sistema. Para un vecindario de **n** termopares se obtienen **n-1** gráficas de correlación, entendiendo que la correlación de una variable con sigo misma siempre es 1.

Se muestran en color azul los valores calculados de la correlación para cada ventana de tiempo. Resulta de interés ver la tedencia de los diferentes valores de correlación, por lo que valores se filtran mediante un filtro de Savitzky-Golay [46] y se muestran superpuestos en color rojo.

Se emplea un filtro Savitzky – Golay ya que, al basarse en un sistema de medias móviles, conserva las tendencias de los valores y suaviza los picos, además de ser un tipo de filtro adecuado para muestras equidistantes a lo largo de la abscisa (el eje temporal en este caso particular) [47].

Se realiza un barrido del ancho de ventana y se comparan los gráficos para buscar cualitativamente un ancho de ventana que permita apreciar la tendencia de los datos en la señal filtrada y la aparición de picos cambios súbitos en las diferentes medidas de correlación. Dado que los datos se capturan con una periodicidad de 15 minutos, se realiza el barrido del ancho de ventana entre los valores presentados en la tabla **5-1**, donde se asocia el tiempo de cada ancho de ventana con la cantidad de registros individuales en dicho intervalo de tiempo.

Ancho de ventana	Cantidad de registros
1 hora	4
6 horas	24
12 horas	48
1 día	96
3 días	288

Tabla 5-1: Anchos de ventana y cantidad equivalente de registros individuales

Además del análisis de correlación en ventanas de tiempo, se estudia la diferencia de temperatura como posible métrica para evaluar el comportamiento de los vecindarios de termopares. Para este propósito se calcula la diferencia de temperatura de cada panel del vecindario respecto un panel de referencia para cada instante de tiempo siguiendo la fórmula de la ecuación (9). El panel de referencia se selecciona en función de qué termopar se quiere estudiar, si el termopar de interés es el panel central del vecindario, este será el panel de referencia para la diferencia de temperatura.

$$\Delta T(t_0)_{panel_i} = T(t_0)_{panel_{ref}} - T(t_0)_{panel_i} \qquad (9)$$

5.2.2. Resultados y discusión

Algunos de los ejemplos estudiados se muestran en las figuras **5-24** a **5-28**. Las figuras corresponden a un vecindario de *Waffle Coolers* donde el panel de referencia es el panel en la posición angular 70 termopar A del cuadrante nor-este, como se muestra en la figura **5-23**. Este mismo tipo de gráficos y análisis se realizaron para cada nivel y tipo de termopar en el horno.

Se calculan los valores de correlación y diferencia de temperatura empleando ancho de ventana de un día, los resultados se muestran entre las figuras **5-24** y **5-28**.

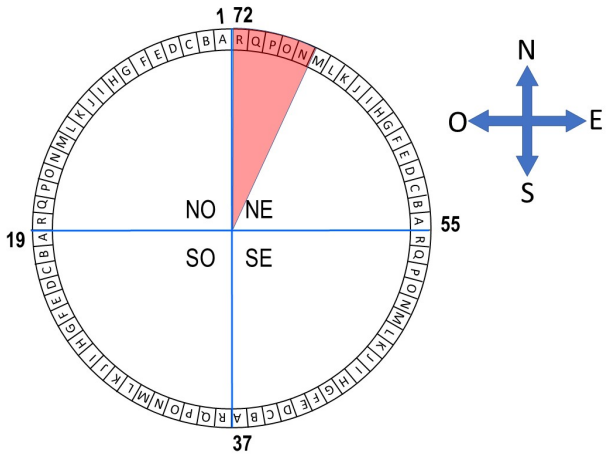


Figura 5-23: Ubicación de termopares de vecindario de *Waffle Coolers*.

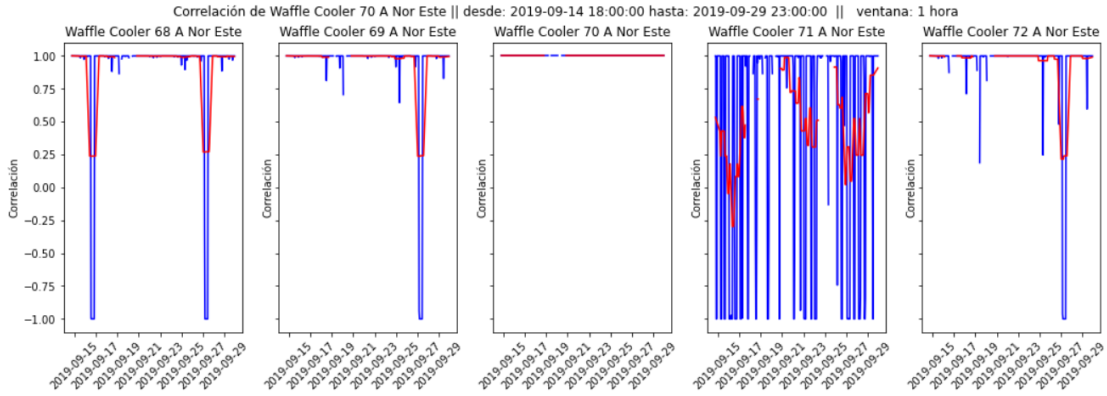


Figura 5-24: Análisis de correlación en ventanas de tiempo de 1 hora

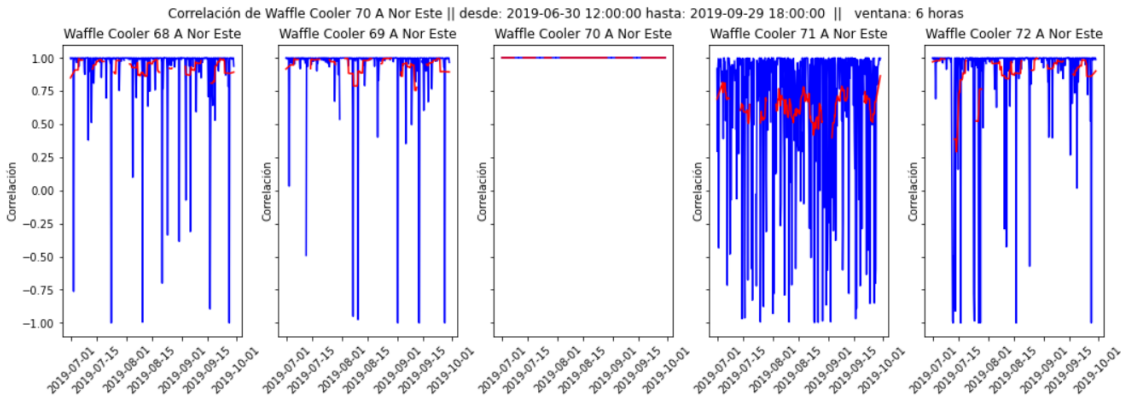


Figura 5-25: Análisis de correlación en ventanas de tiempo de 6 horas

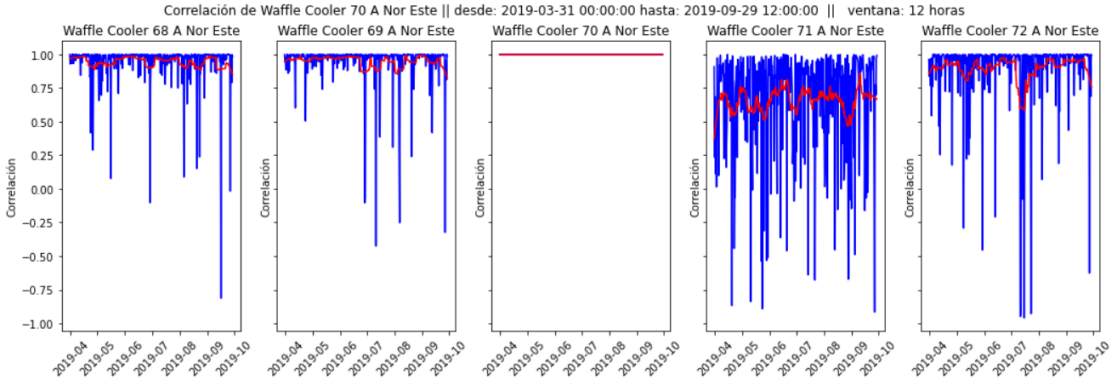


Figura 5-26: Análisis de correlación en ventanas de tiempo de 12 horas

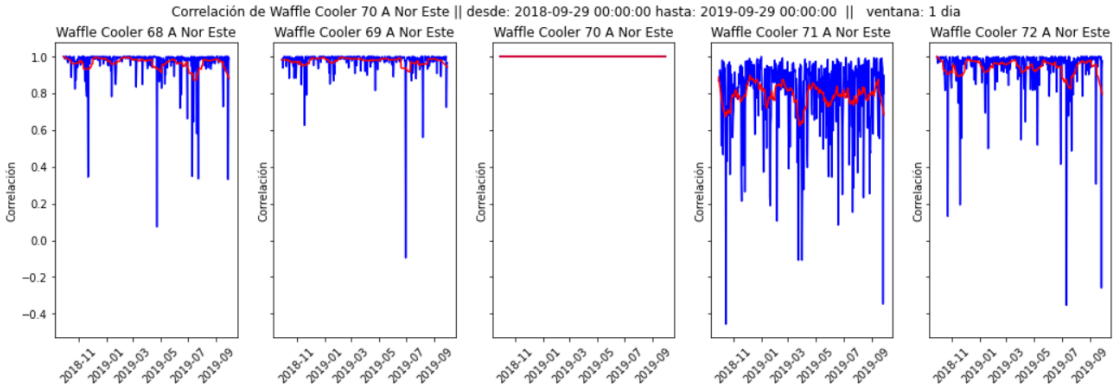


Figura 5-27: Análisis de correlación en ventanas de tiempo de 1 día

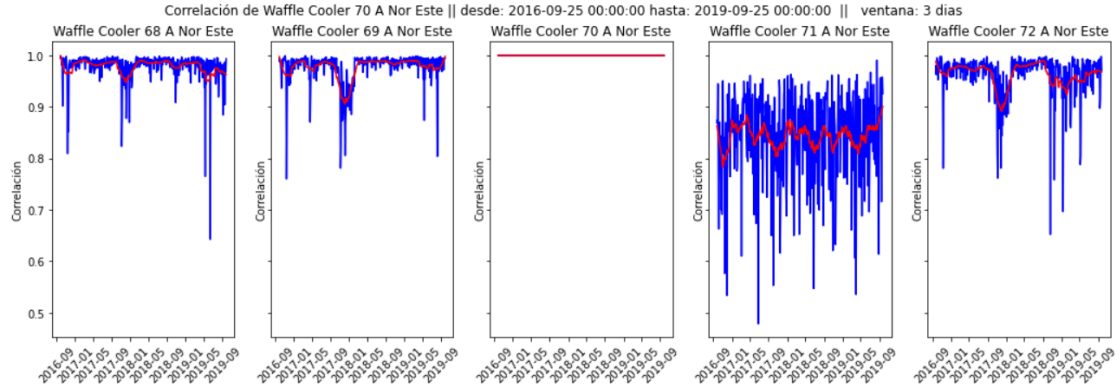


Figura 5-28: Análisis de correlación en ventanas de tiempo de 3 días

Se evidencia que la correlación entre cada variable y el panel de referencia varía con el tiempo y que el cambio en el ancho de la ventana de tiempo tiene incidencia en la cantidad y magnitud de picos identificables en las diferentes gráficas de correlación. La selección del ancho de ventana adecuado permite balancear entre la sensibilidad que se consigue mediante ventanas de tiempo pequeñas y la capacidad de inferencia de las tendencias que se consigue seleccionando ventanas de tiempo anchas.

Del análisis resultan de especial interés las ventanas de tiempo con ancho de 6 horas y de 1 día, las primeras por la sensibilidad a los cambios en la correlación y las segundas por la facilidad con la que se aprecia la tendencia de los datos.

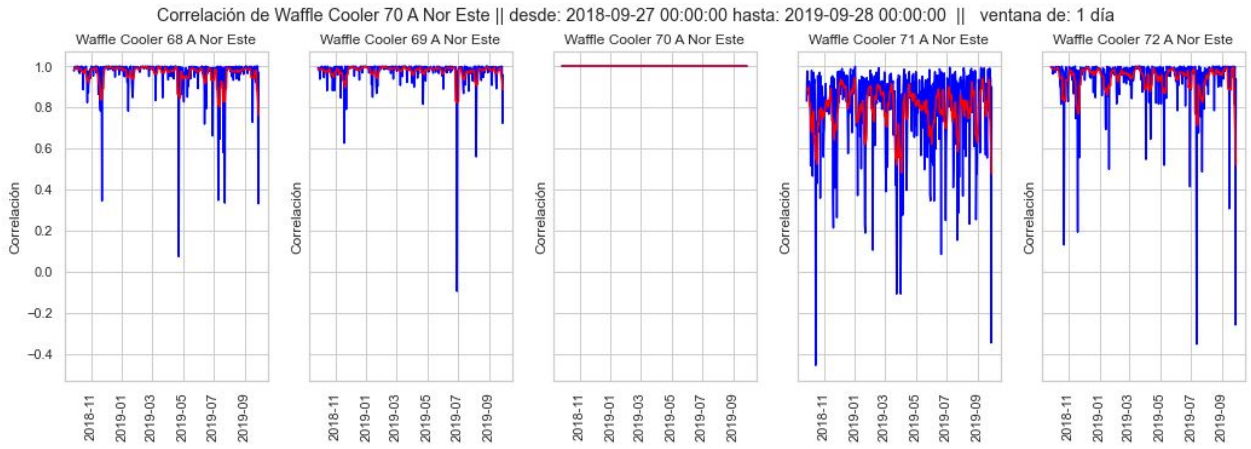


Figura 5-29: Análisis de correlación en vecindario de *Waffle Coolers*.

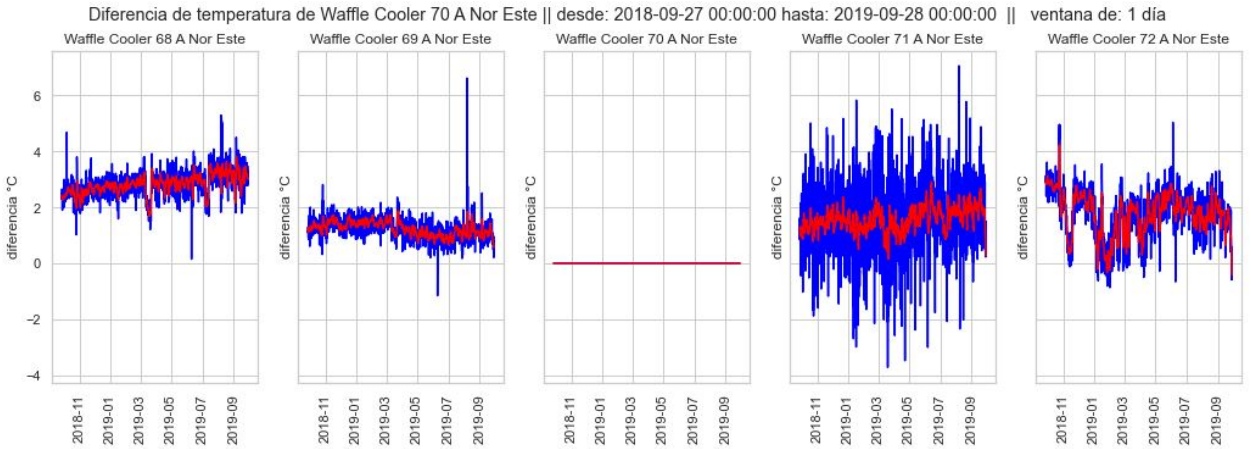


Figura 5-30: Diferencia de temperatura en vecindario de *Waffle Coolers*.

Se evidencia tanto la existencia de picos de baja correlación, así como la existencia de picos

en la diferencia de temperatura. A partir de la observación de los picos presentes en las figuras **5-29** y **5-30**, especialmente las gráficas correspondientes a los *Waffle Coolers* 68 A y 69 A, se observa que no todo pico de baja correlación está acompañado de un pico en la diferencia de temperatura.

Se toma como primer caso el pico de baja correlación en el *Waffle Cooler* 69 A justo antes de la marca de tiempo 2019–07. Dicho pico de baja correlación no se encuentra acompañado por un pico (ni positivo ni negativo) de diferencia de temperatura, lo que nos indica que la baja correlación entre dos variables puede deberse a factores diferentes a la existencia de picos en la diferencia de temperatura entre los dos termopares. La figura **5-31** muestra las medidas de temperatura de los *Waffle Coolers* 69 A y 70 A alrededor del inicio del mes de julio de 2019.

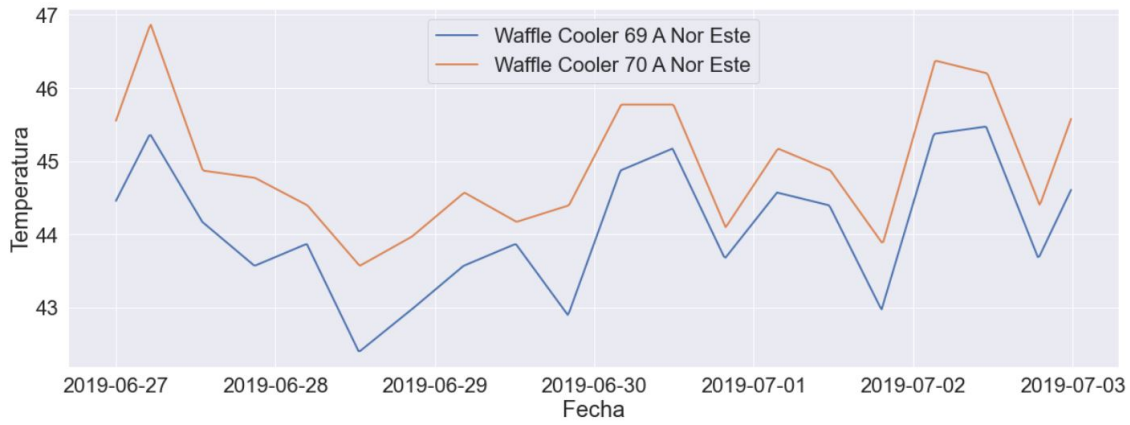


Figura 5-31: Temperatura de *Waffle Coolers* 69 A y 70 A alrededor del 01-07-2019.

Se observa que el 29 de julio de 2019, hacia la segunda mitad de dicho día, ocurre un fenómeno en el cual la temperatura del *Waffle Cooler* 69 A crece a una tasa constante, mientras que la temperatura del *Waffle Cooler* 70 A desciende y vuelve a subir posteriormente. Esta divergencia transitoria entre las temperaturas de los dos termopares vecinos ocasiona que la medida de correlación caiga en la ventana de tiempo en que sucede este fenómeno. Se observa también que, posterior a la ocurrencia de este fenómeno, los termopares vuelven a presentar el comportamiento esperado para dos termopares vecinos.

La presencia ventanas de tiempo con picos de baja correlación entre las señales de pares miembros de un vecindario, coincidentes con la aparición de anomalías en el comportamiento de al menos uno de los termopares del vecindario, es un fenómeno que también se evidencia en en los *Plate Coolers* de un mismo nivel.

También se observa que la aparición de un pico en la diferencia de temperatura no necesariamente va acompañada de un pico de baja correlación entre las señales de los *Plate Coolers*.

5.3. Detección heurística de valores anclados en ventanas de tiempo

Dentro de los posibles fallos abruptos en un termopar, los fallos conocidos como:

- Circuito abierto
- Corto circuito
- LSB congelado en 1
- LSB congelado en 0

Estos fallos tienen en común que el valor leído del sensor se estanca en un valor determinado y permanece invariante hasta que ocurra algún cambio en el sistema de medición. Los fallos por circuito abierto se caracterizan porque el valor leído del sensor es el máximo posible dado que no existe flujo de corriente entre el sensor y su transductor; los fallos por cortocircuito generan lecturas de valor 0 al circular la corriente máxima en el circuito de medición; los fallos por congelamiento en valor del bit menos significativo generan lecturas invariantes del último valor conocido de la variable con un valor que depende de la interpretación de la variable leída.

Cualquiera sea el caso, al observar las lecturas en el tiempo de estos tipos de fallos, la invarianza del valor leído ocasiona que al calcular la varianza de los valores leídos en un intervalo de tiempo, este valor sea 0 independiente del valor en que la variable se encuentre anclada. Esto permite detectar rápidamente este tipo de fallos al realizar el cálculo de la varianza de la señal en una ventana de tiempo suficientemente pequeña.

5.3.1. Metodología

Se emplea un método heurístico para la detección de valores anclados a partir de la varianza de la señal en ventanas de tiempo. La figura **5-32** muestra el proceso para obtener una métrica para detección de valores anclados en una línea de tiempo.

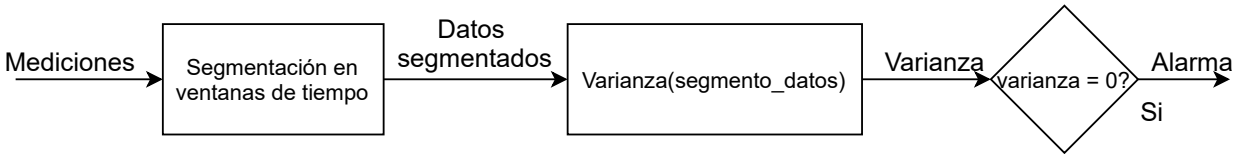


Figura 5-32: Detección de datos anclados por varianza

5.3.2. Resultados y discusión

Al emplear ventanas de ancho de 1 día se identificaron 35 termopares con datos congelados en diferentes ventanas de tiempo. La tabla **5-2** muestra los termopares identificados con datos anclados y la cantidad de ventanas identificadas para uno. En las tablas se abrevian los nombres de los *Plate Coolers* como *Plate* y los nombres de los *Waffle Coolers* como *Waffle*.

Termopar	Vent. Ancladas
Waffle 34 B	1435
Waffle 58 C	1277
Waffle 42 B	556
Waffle 58 D	262
Waffle 35 A	241
Plate 44 D	200
Plate 35 A	107
Plate 43 A	103
Plate 43 D	54
Plate 34 A	48
Plate 34 C	48
Waffle 50 B	37
Plate 52 C	19
Plate 56 A	19
Plate 35 D	16
Plate 36 A	16
Plate 30 A	15
Plate 30 C	14

Termopar	Vent. Ancladas
Waffle 54 B	11
Waffle 53 A	11
Waffle 53 B	11
Waffle 33 A	11
Plate 31 D	10
Waffle 56 B	10
Waffle 56 A	10
Waffle 55 B	10
Waffle 55 A	10
Waffle 54 A	10
Waffle 50 A	10
Waffle 49 B	10
Waffle 49 A	10
Plate 56 C	10
Waffle 59 A	10
Plate 52 A	6
Plate 44 A	3

Tabla 5-2: Conteo de ventanas congeladas de ancho 1 día para termopares del horno

Se selecciona el *Waffle Cooler* 56 B y se observan las ventanas de tiempo en que se detectaron valores anclados. Los resultados indican que los datos anclados se encuentran entre las fechas 2015-03-07 y 2015-03-16. Se genera una gráfica en el tiempo de los valores de dicha variable alrededor de las fechas indicadas, el resultado se muestra en la figura **5-33**.

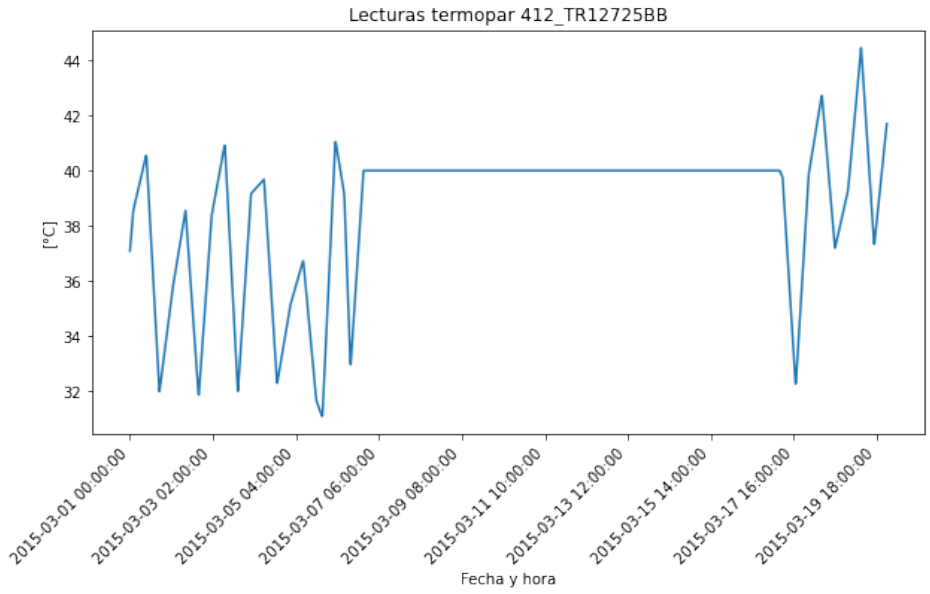


Figura 5-33: Lecturas del *Waffle Cooler* 56 B alrededor de fechas detectadas

El algoritmo detecta efectivamente el intervalo de tiempo en que el valor de la variable se queda anclado en el valor 40 °C. Resultados análogos se encuentran en las demás variables identificadas de la tabla **5-2**. El algoritmo detecta un total de 4630 ventanas individuales, que equivalen al 7.874 % del total de los datos existentes en el dataset. La tabla **5-3** muestra una comparación de la detección de ventanas con valores anclados variando el ancho de la ventana.

Ancho ventana	Ventanas dataset	Ventanas ancladas	Porc. con fallo
1 hora	14'111,328	283,031	2.0057
6 horas	2'351,888	24,620	1.0468
12 horas	1'175,944	9,978	0.8485
1 día	587,972	4,690	0.7976
1 semana	83,720	604	0.7214

Tabla 5-3: Comparación de detección de valores anclados con ventanas de tiempo de diferente ancho

Se observa un incremento considerable en el porcentaje de ventanas identificadas como ancladas cuando se emplea ancho de ventana de 1 hora respecto a los demás casos evaluados. No se emplean anchos de ventana menores a 1 horas debido a que los datos se registran con una tasa de muestreo de 15 minutos.

El incremento en el porcentaje de datos anclados identificados al emplear ventanas de ancho de 1 hora puede atribuirse a la aparición de fallos intermitentes, donde el fallo aparece por un intervalo de tiempo corto y posteriormente se restablece el funcionamiento normal del sensor. Debido a que el sistema de medición de temperatura registra el valor con resolución de 6 cifras decimales, no se considera la posibilidad de que la falta de cambio en el valor medido se deba a falta de cambio en la temperatura real del proceso.

5.4. Detección heurística de outliers

5.4.1. Metodología

Para el proceso de detección de outliers se propone un proceso de detección en ventanas de tiempo ajustables donde se cuenta la cantidad de outliers presente en la ventana y se establece un segundo parámetro ajustable que es el umbral de cantidad de outliers admisible antes de levantar la alarma de presencia de outliers.

La figura **5-34** ilustra el proceso que se sigue en la determinación del estado de la alarma de outliers.

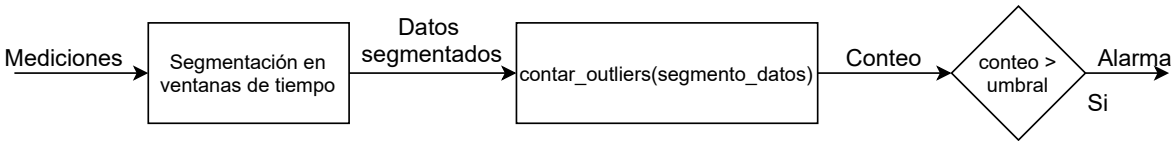


Figura 5-34: Detección de outliers en ventanas de tiempo

Se implementa el algoritmo descrito en la figura **5-34** con la posibilidad de configurar el ancho de ventana y el umbral de outliers. El conteo se realiza para cada variable de manera individual y la etapa de conteo de outliers genera un dataframe en ventanas de tiempo con el conteo de outliers en cada ventana, que después es comparado con el umbral para decidir si levanta o no la alarma.

5.4.2. Resultados y discusión

Se ejecuta el algoritmo de detección de outliers con ancho de ventana de 1 día y umbral de 5 outliers por ventana para levantar la alarma, considerando como outliers valores por fuera del intervalo $[20, 100]$ °C indicado en documentos técnicos [1]. Se identifica un total de 91 variables con cantidad de outliers por encima del umbral. La tabla **5-4** muestra algunas de las variables con los mayores conteos de outliers detectados por el algoritmo y la variable con menor conteo de ventanas detectadas con outliers.

La consideración inicial de 5 outliers como umbral de alarma de outliers se debe a que un día de operación produce 96 registros con una periodicidad de 15 minutos. Considerar como umbral de alarma 5 registros de 96 existentes, es equivalente a decir se levanta la alarma si el porcentaje de registros fuera del intervalo de operación es superior al 4.8 %.

Termopar	**Vent. Outliers**	**Termopar**	**Vent. Outliers**
Waffle 42 B	514	Waffle 58 C	122
Waffle 35 A	284	Plate 35 D	122
Plate 34 D	254	Plate 28 D	117
Plate 32 D	246	Plate 24 D	117
Plate 36 D	201	Plate 12 D	117
Plate 14 D	177	Plate 6 D	115
Plate 33 D	177	Waffle 33 A	111
Plate 30 D	169	Plate 22 D	105
Plate 31 D	156	Plate 8 D	101
Plate 10 D	127	Waffle 29 B	1

Tabla 5-4: Conteo de ventanas con outliers, ventana 1 día y umbral 5 para termopares del horno

Se toma como caso de comprobación el *Waffle Cooler* 29 B y se observa la ventana en que se identificó la presencia de outliers. La ventana que inicia en la fecha 2017-12-13, mostrada en la figura **5-35**, muestra el comportamiento de los datos de este termopar alrededor de la ventana de interés.

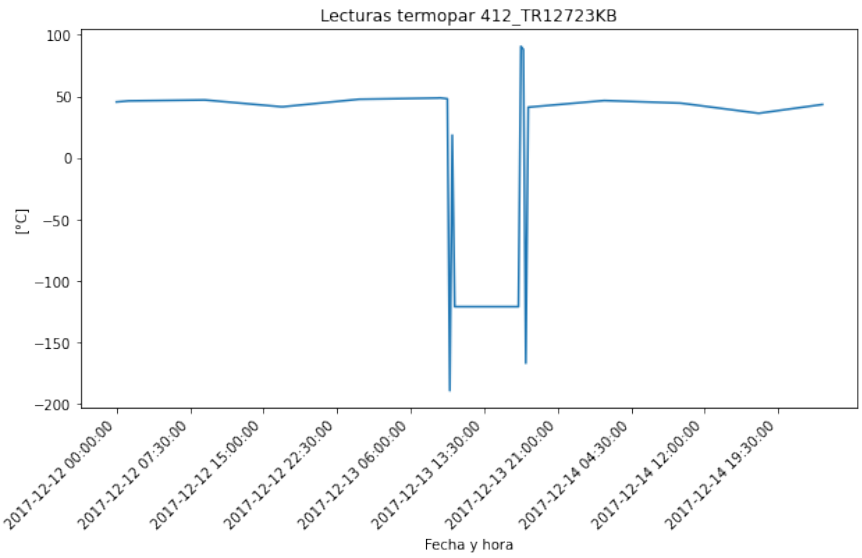


Figura 5-35: Comportamiento de outliers del *Waffle Cooler* 29 B alrededor del intervalo de interés

Se aprecia la existencia de outliers en el intervalo detectado por el algoritmo. La capacidad de detección se muestra consistente en las demás variables y ventanas identificadas por el algoritmo. Se realiza un barrido de parámetros tanto del ancho de ventana como del porcentaje de umbral de alarma. Los resultados se muestran en la tabla **5-6**. El porcentaje que se muestra en la tabla se calcula mediante la ecuación (10).

$$P_o = 100 * \frac{\sum_{i=1}^{n} Cant.Ventanas(outliers >= umbral)_{Termopar_i}}{Cant.Termopares * Total.Ventanas}[\,\%] \qquad (\ 10\ )$$

Para todos los casos la cantidad de termopares es 322. Los anchos de ventana y total de ventanas en que se divide el dataset en cada caso, se relacionan en la tabla **5-5**.

Ancho ventana	Registros por ventana	Total ventanas
1 hora	4	14'111,328
6 horas	24	2'351,888
12 horas	48	1'175,944
1 día	96	587,972
1 semana	672	83,720

Tabla 5-5: Caracterización de ventanas de tiempo en dataset

En la tabla **5-6** se muestran los resultados de un ejercicio de conteo del total de porcentaje de ventanas con outliers respecto al total de ventanas haciendo un barrido de dos parámetros: umbral de porcentaje de outliers y ancho de ventana (tiempo). En los casos en que el umbral de registros con outliers necesarios para etiquetar la ventana como con outliers no es un número entero se aplica la función techo para garantizar que el umbral mínimo sea de 1 registro por ventana.

Se notará en la tabla **5-6** que existen casos en que se combinan valores de umbrales de detección, esto se debe a que ventanas de menor ancho presentan menos registros por ventana y esto ocasiona que algunos valores de umbral no sean aplicables dado que el valor del umbral en cantidad de registros será, en ocasiones, el mismo para dos o más casos.

	Umbral de porcentaje de outliers					
Ancho ventana	**5**	**10**	**20**	**30**	**40**	**50**
1 hora	0.5051					
6 horas	0.6191	0.6040	0.5777	0.5401	0.5180	0.4942
12 horas	0.7161	0.6862	0.6205	0.5631	0.5167	0.4821
1 día	0.8730	0.7981	0.6806	0.6022	0.5217	0.4522
1 semana	1.3760	1.0547	0.7501	0.5554	0.4264	0.3225

	60	**70**	**80**	**90**	**100**
1 hora	0.4767				
6 horas	0.4645	0.4457	0.4157	0.3907	0.3775
12 horas	0.4444	0.4078	0.3712	0.3344	0.3058
1 día	0.3950	0.3459	0.3027	0.2637	0.2321
1 semana	0.2568	0.2114	0.1612	0.1194	0.0991

Tabla 5-6: Comparación de detección de ventanas con outliers en ventanas de tiempo de diferente ancho

El comportamiento de los valores mostrados en la tabla **5-6** se muestra gráficamente en la figura **5-36**.

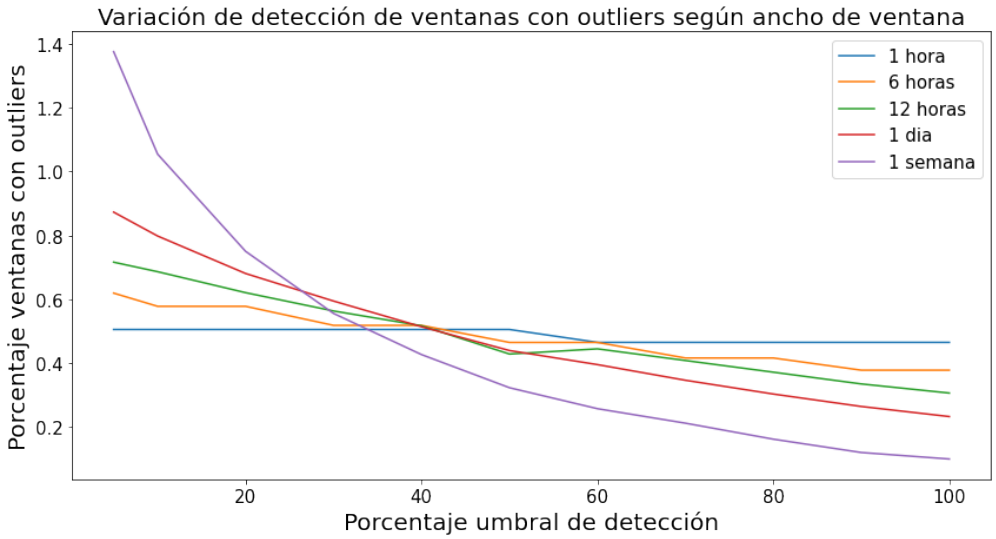


Figura 5-36: Variación de detección de ventanas con outliers según ancho de ventana

Se observa que el porcentaje de ventanas detectadas incrementa de manera inversa respecto a la magnitud del umbral de porcentaje de outliers, esto se debe a que entre menor sea el valor del umbral, será menor también la cantidad mínima de outliers individuales que deben aparecer para levantar la bandera de alarma.

5.5. Función de Distribución Acumulativa (CDF) y presencia de fallos

Mandal *et. al* [48] presentaron un método para la detección de fallos menores en termopares con base en la función de disitribución acumulativa o *CDF* por su sigla en inglés. La *CDF* se puede definir como la suma acumulativa de la función distribución de probabilidad. En la figura **5-37**, se muestra una distribución normal estándar y su correspondiente *CDF*.

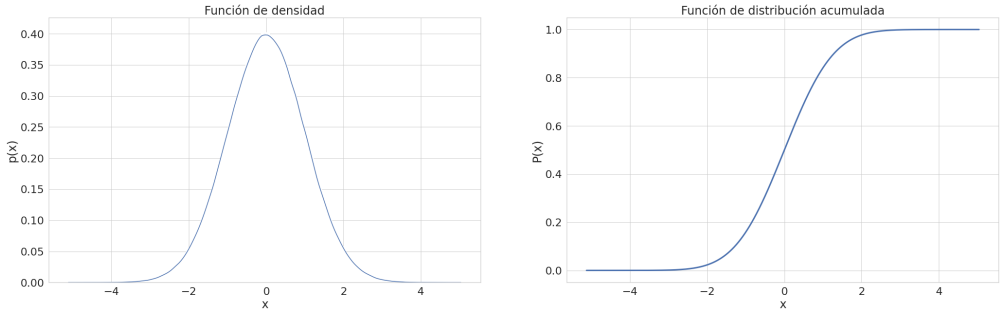


Figura 5-37: Función de distribución de probabilidad y *CDF* para datos con distribución normal estándar

El trabajo de estos autores se basa en el teorema de Glivenko-Cantelli, el cual establece que la *CDF* empírica de variables aleatorias independientes suficientemente grandes convergen hacia la *CDF* verdadera [49].

Mandal *et al* construyen la *CDF* característica de los sensores a partir de datos conocidos de los termopares etiquetados como sin fallo, a llamarse *CDF* de referencia en próximas menciones, y comparan repetidas veces *CDF*s de diferentes conjuntos de datos (sin fallos, datos con deriva, datos anclados) con la *CDF* de referencia para determinar la existencia de fallos a partir de la distancia entre las dos distribuciones. La distancia entre dos *CDF* la definen los autores como la máxima distancia en la ordenada que separa las *CDF*s de referencia y estudiada. En el ejemplo de la figura **5-38** se observa que la máxima distancia entre las dos distribuciones, se encuentra en un punto mayor que cero en la abscisa, marcado con color rojo y encerrado en el rectángulo verde.

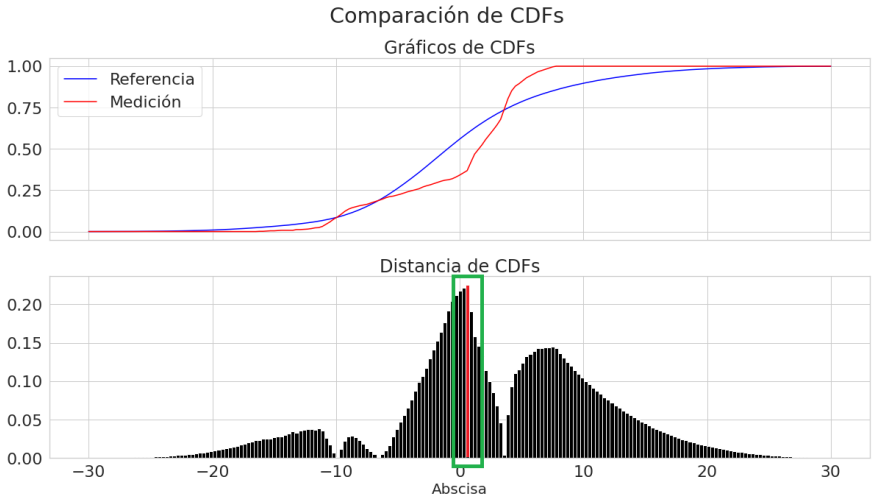


Figura 5-38: Ejemplo de comparación de dos *CDF*s y medida de la distancia entre las distribuciones

Los autores utilizan los datos de las comparaciones de las *CDF*s de diferentes conjuntos de datos y la *CDF* de referencia para determinar un valor umbral de diferencia entre las *CDF*s para determinar la existencia de un fallo. Si la distancia entre la *CDF* bajo estudio y la *CDF* de referencia supera el valor umbral, los autores determinan que existe un fallo en el sensor.

5.5.1. Metodología

De los métodos desarrollados hasta este punto en el proyecto, la detección heurística de outliers y de ventanas ancladas permiten detectar y aislar efectivamente estos dos tipos comunes de fallos en termopares individuales.

Para establecer *CDF*s de referencia para la red de termopares del horno se lleva a cabo un proceso de detección, aislamiento y eliminación de ventanas con presencia de outliers y ventanas ancladas, siguiendo los procesos mostrados en las secciones 5.3 y 5.4. Se emplea el análisis en ventanas de tiempo y, con base en el teorema de Glivenko-Cantelli, se clasifican y agrupan los termopares según su tipo y cuadrante para calcular las *CDF* de referencia empíricas siguiendo el proceso mostrado en la figura **5-39**. Para efectos del diagrama, la sigla *D.A.E* hace referencia al proceso de detección, aislamiento y eliminación de datos identificados con algún tipo de fallos. Los procesos resaltados en color verde y encerrados en un cuadro se realizan en ventanas de tiempo, mientras que los demás procesos se realizan una única vez para cada panel.

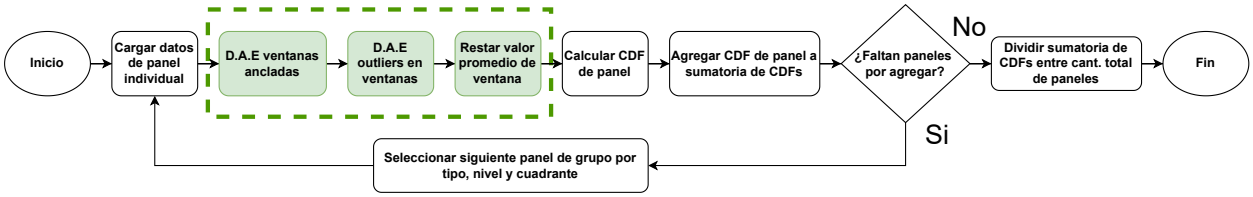


Figura 5-39: Diagrama de flujo de cálculo de *CDF* por tipo, nivel y cuadrante de panel

El proceso de restar el valor promedio de la ventana de tiempo tiene por objeto eliminar la incidencia del valor D.C. de la señal y obtener la *CDF* únicamente de la dispersión de los valores registrados para los diferentes paneles. Se emplea una combinación de anchos de ventana de 1 día para la detección de outliers y de 1 hora para la detección de ventanas ancladas, dichas ventanas son eliminadas de la línea de tiempo para el cálculo de la *CDF* de cada panel.

Una vez obtenidas las *CDF* de referencia empíricas para los diferentes tipos, niveles y cuadrantes de paneles, se realizan ejercicios de comparación entre las *CDF* de referencia de los paneles y las *CDF* de intervalos de tiempo identificados tanto con como fallos como sin fallos, con el fin de determinar si es posible establecer un valor umbral para identificar fallos en termopares a partir del análisis de *CDF*.

5.5.2. Resultados y discusión

En las figuras **5-40** a **5-44** se muestran las *CDF* obtenidas para los *Plate Coolers* de niveles A hasta D y para los *Waffle Coolers*, respectivamente.

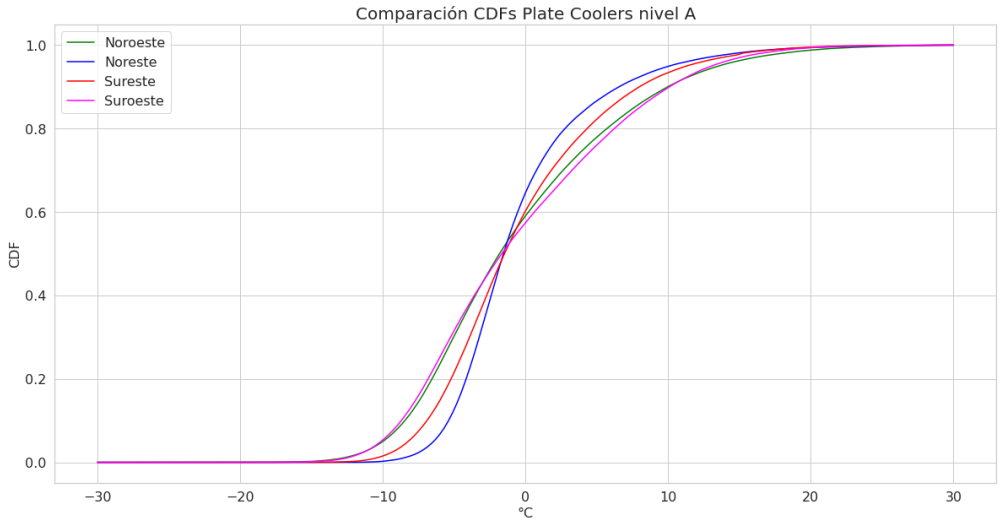


Figura 5-40: Comparación de *CDF*s de *Plate Coolers* de nivel A para los 4 cuadrantes del horno

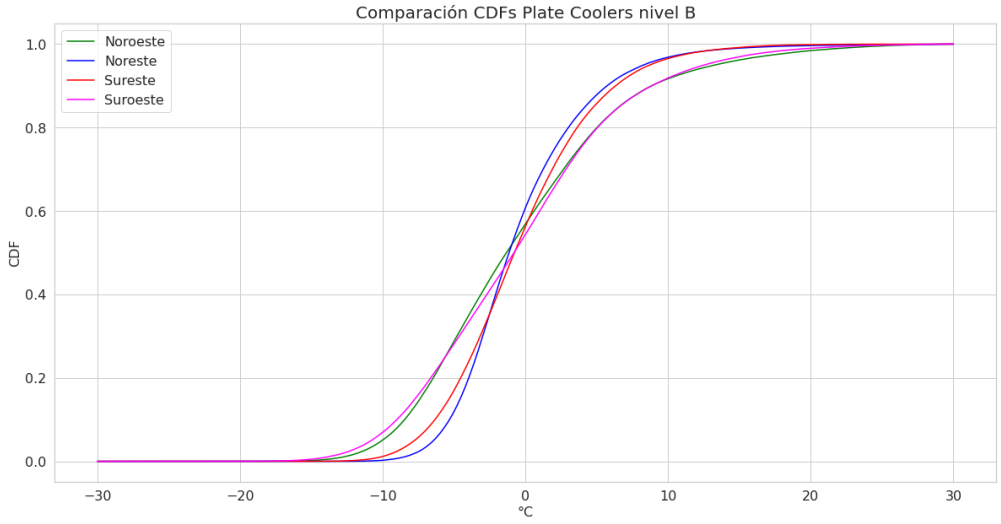


Figura 5-41: Comparación de *CDF*s de *Plate Coolers* de nivel B para los 4 cuadrantes del horno

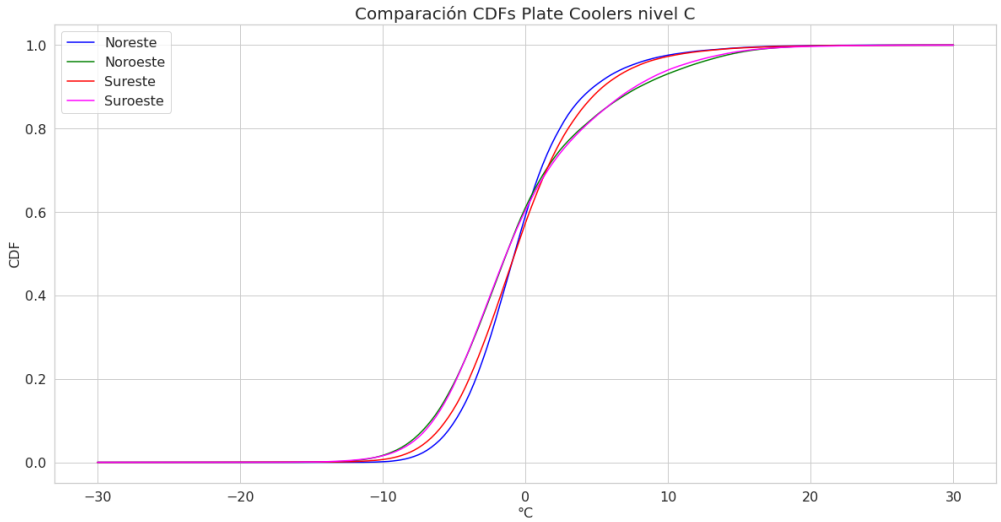


Figura 5-42: Comparación de *CDF*s de *Plate Coolers* de nivel C para los 4 cuadrantes del horno

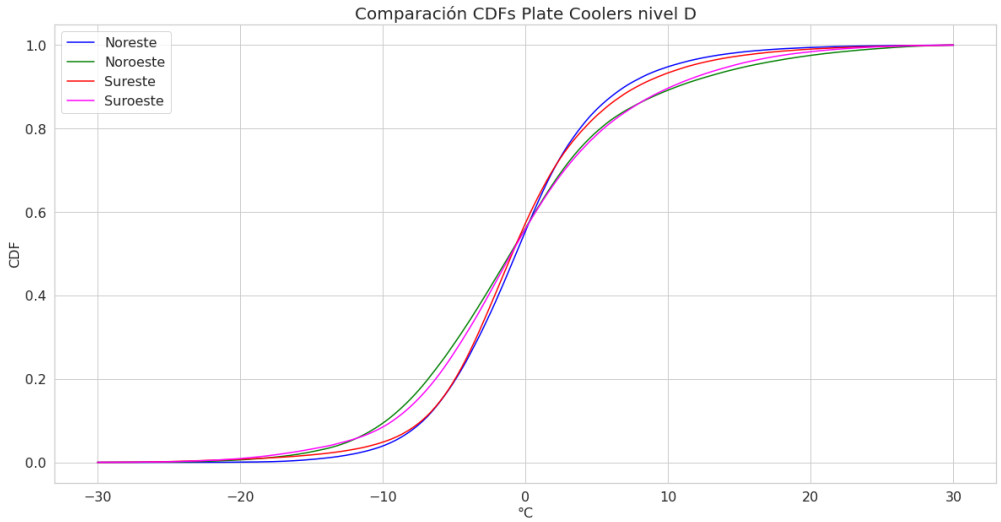


Figura 5-43: Comparación de *CDF*s de *Plate Coolers* de nivel D para los 4 cuadrantes del horno

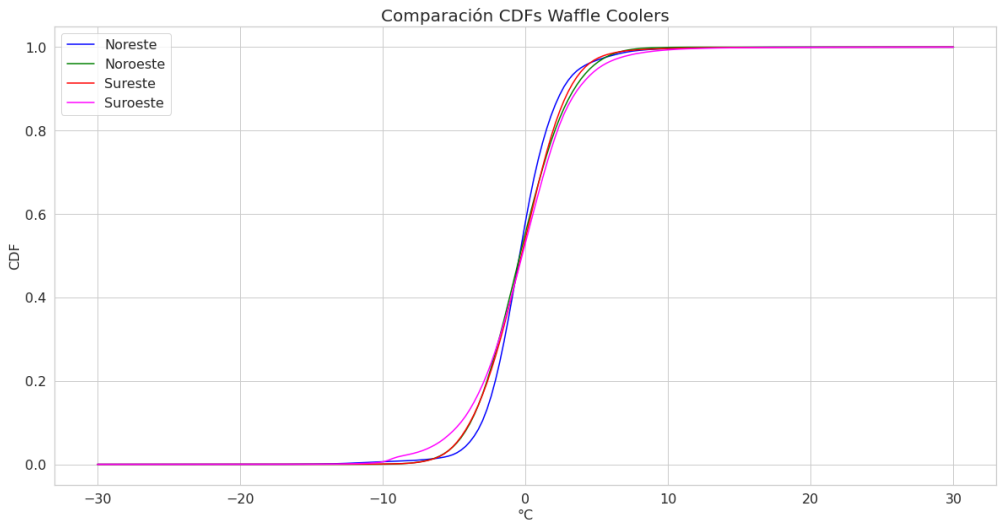


Figura 5-44: Comparación de *CDF*s de *Waffle Coolers* para los 4 cuadrantes del horno

De las 5 figuras se observa que las *CDF* de los *Plate Coolers* en los niveles A y B presentan las mayores diferencias de temperatura entre los 4 cuadrantes del horno para un mismo nivel. Una posible razón de este fenómeno es que estos son los termopares de los dos niveles superiores del horno y en ellos se deposita la calcina para iniciar gradualmente su proceso de fundición y posterior separación en metal fundido y escoria.

Dado que el ingreso de calcina al horno depende en su totalidad de la acción manual de los operarios bajo su criterio experto, la temperatura en los niveles superiores del horno es más propensa a experimentar mayores variaciones que en los niveles inferiores del horno donde la temperatura del material que reacciona depende principalmente de la propiedades fisicoquímicas de la calcina.

5.5.3. Distancia entre CDFs para termopares del horno

Se emplea información obtenida de las bitácoras de mantenimiento para seleccionar termopares y fechas en los que se tenga mayor certeza de la existencia y posterior correción de un estado de fallo, se calculan las *CDF* de las medidas antes y después del mantenimiento y se comparan con la *CDF* de referencia para estudiar cambios en la distancia máxima entre las *CDF* de los datos con fallo (antes del mantenimiento) y libres de fallo (después del mantenimiento).

Convención para nombrar los termopares: A partir de este punto se mencionan los termopares a partir de su tipo, posición angular y nivel. De tal forma, si se menciona el *Plate Cooler* 12 C se está haciendo referencia al *Plate Cooler* en la posición angular 12 del nivel C.

El primer caso que se estudia es el *Plate Cooler* 35 D, del cual se establece en la bitácora que presentaba fallos hasta la fecha 2015-04-05, fechan en la que se le realizó mantenimiento. La figura **5-45** muestra los datos almacenados tanto para el *Plate Cooler* 35 D como para dos paneles adyacentes en el nivel D, los *Plate Coolers* 34 D y 36 D, se señala en color rojo la fecha en la que se indica que se realizó el mantenimiento.

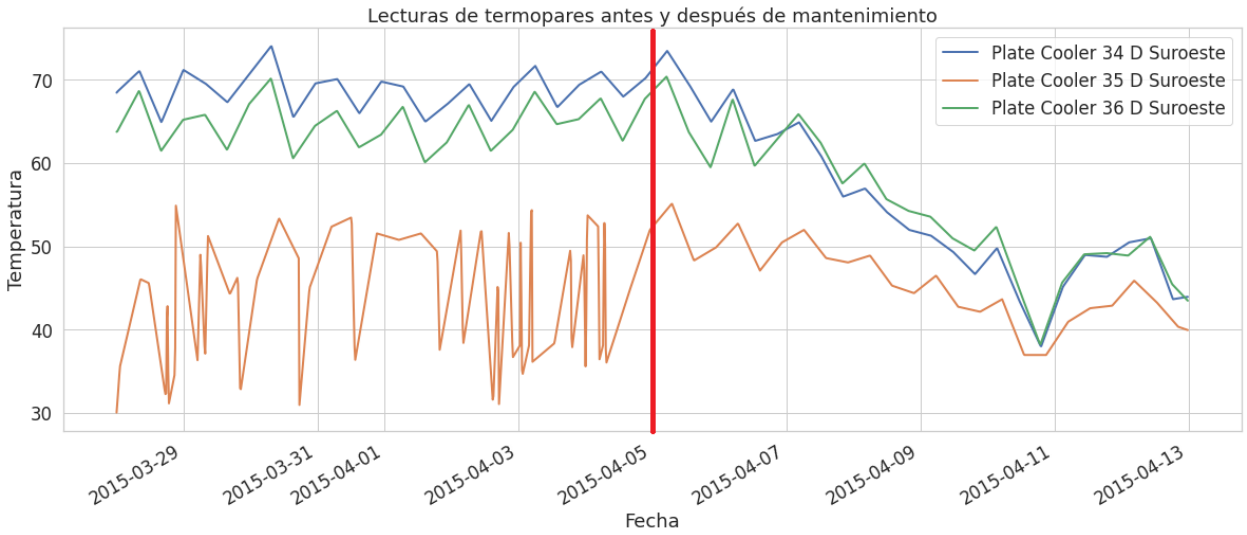


Figura 5-45: *Plate Coolers* 34 D a 36 D antes y después de mantenimiento

Se observa en la figura **5-45** que las medidas obtenidas del *Plate Cooler* 35 D antes del mantenimiento presentan un comportamiento marcadamente diferente respecto al de sus termopares vecinos. Se evidencia también que, después de la fecha registrada de mantenimiento del *Plate Cooler* 35 D, las medidas registradas de los tres termopares vecinos varían con mayor similitud entre los tres sensores.

Se calcula la *CDF* del *Plate Cooler* 35 D antes del mantenimiento, se calcula la distancia máxima respecto a la *CDF* de referencia para los *Plate Coolers* del cuadrante suroeste y se repite el proceso para los datos registrados después del mantenimiento. Considerando que no se especifica en la bitácora la hora de inicio y fin del mantenimiento, no se incluyen en los cálculos ningún dato del día 2015-04-05. Las figuras **5-46** y **5-47** muestran las *CDF*s y las distancias respecto a la referencia para los datos registrados del *Plate Cooler* 35 D antes y después del mantenimiento, respectivamente.

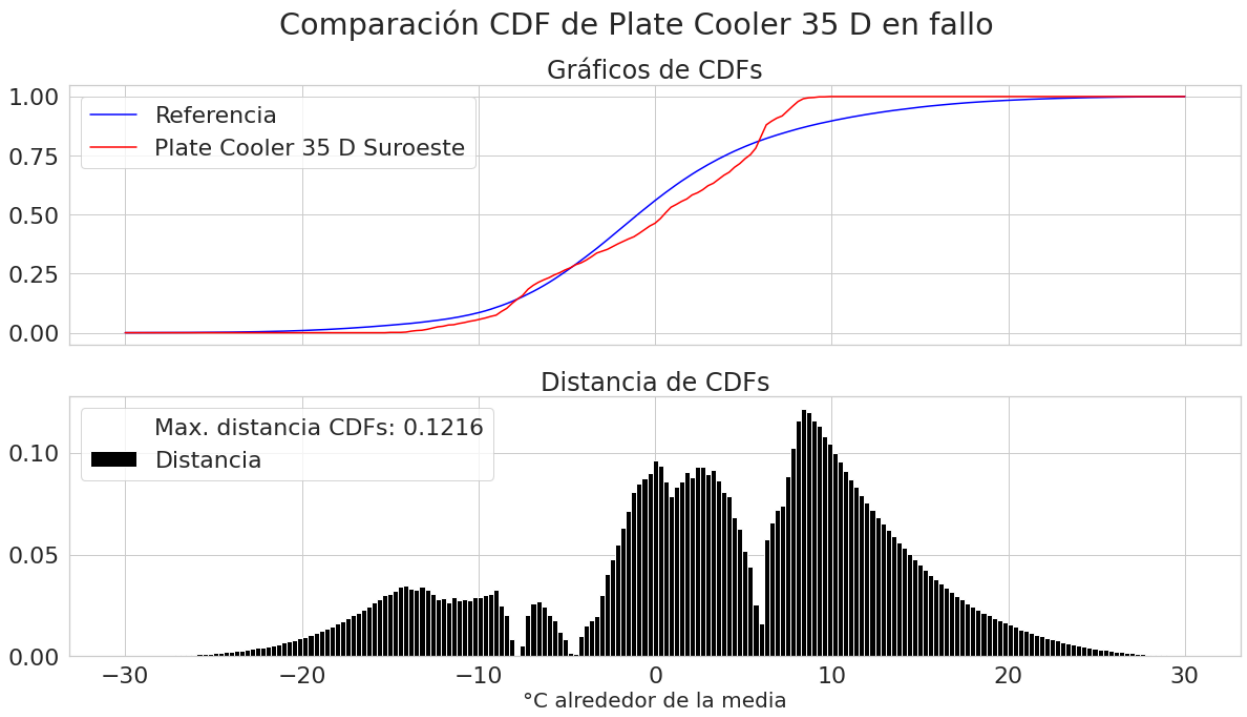


Figura 5-46: *CDF* del *Plate Cooler* 35 D antes del mantenimiento

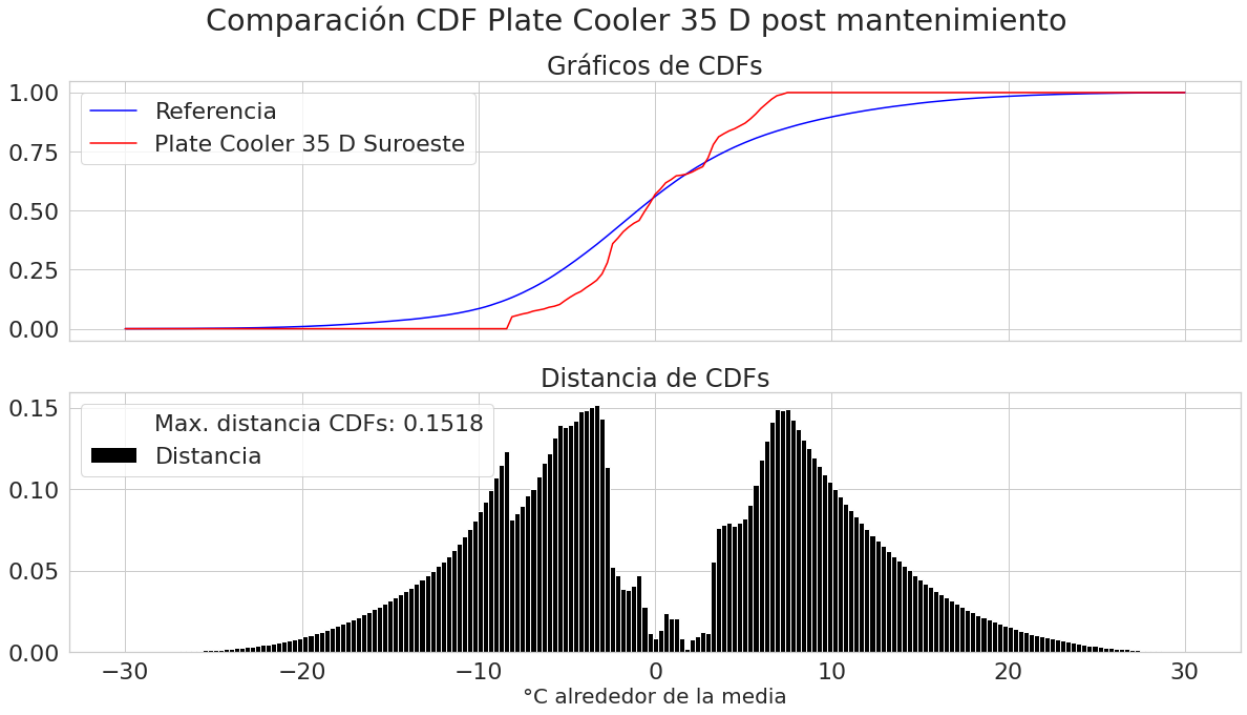


Figura 5-47: *CDF* del *Plate Cooler* 35 D después del mantenimiento

En el trabajo desarrollado por el autor existe una notable diferencia entre la distancia de las *CDF* con y sin fallos respecto a la referencia, por lo que se procede a evaluar el comportamiento de las *CDF* y las distancias máximas en los termopares adyacentes al *Plate Cooler* 35 D para los mismos intervalos de tiempo. Se hace énfasis en que los *Plate Coolers* 34 D y 36 D se encuentran libres de fallos desde antes del mantenimiento, de acuerdo con la bitácora y con lo observado en la figura **5-45**. En las figuras **5-48** y **5-49** se muestran las comparaciones de *CDF*s para los *Plate Coolers* 34 D y 36 D antes del mantenimiento, respectivamente.

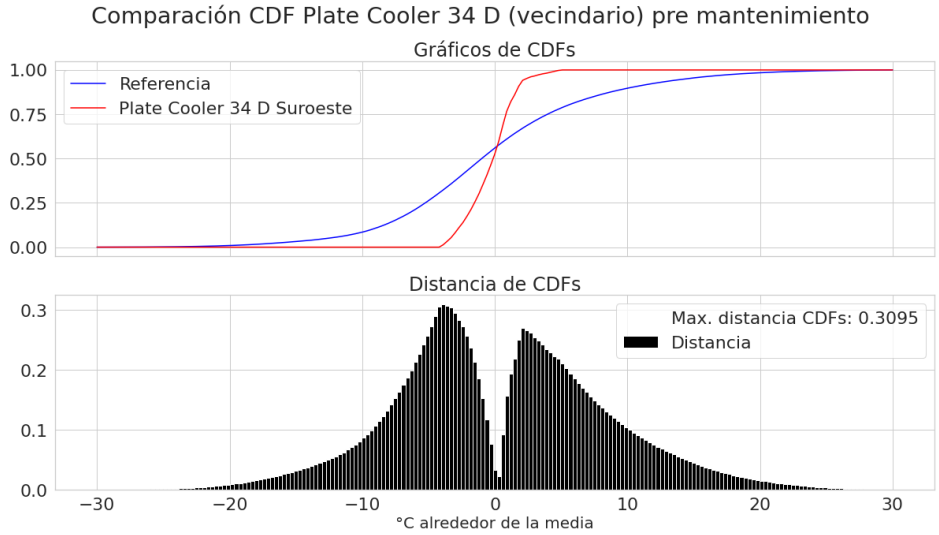


Figura 5-48: *CDF* del *Plate Cooler* 34 D antes del mantenimiento

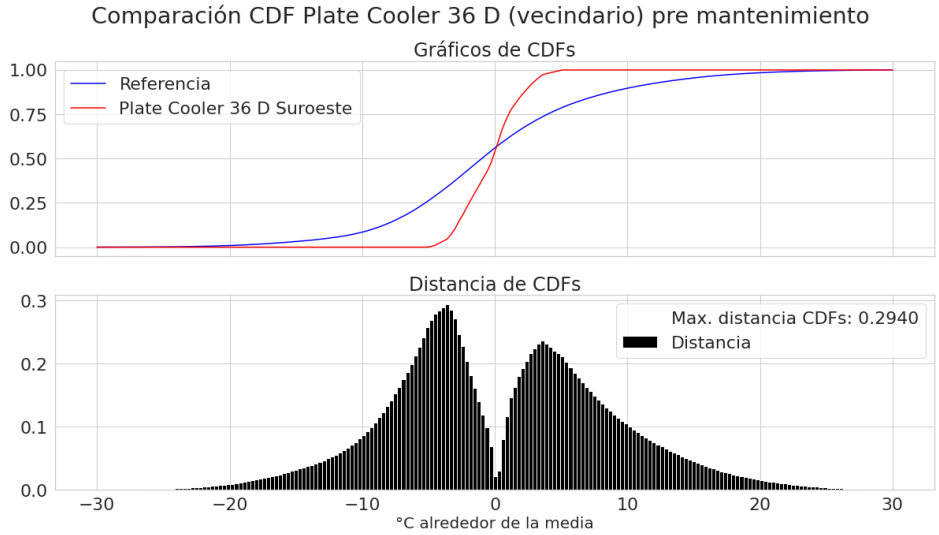


Figura 5-49: *CDF* del *Plate Cooler* 36 D antes del mantenimiento

Se observa que en ambos casos la máxima distancia entre las *CDF* de los datos medidos y la *CDF* de referencia se encuentra alrededor de 0.3, valor que es más del doble que la distancia máxima de *CDF* registrada en el *Plate Cooler* 35 D antes del mantenimiento. A continuación, se muestra en las figuras **5-50** y **5-51** el estudio de *CDF*s para los *Plate Coolers* 34 D y 36 D, respectivamente, después del mantenimiento.

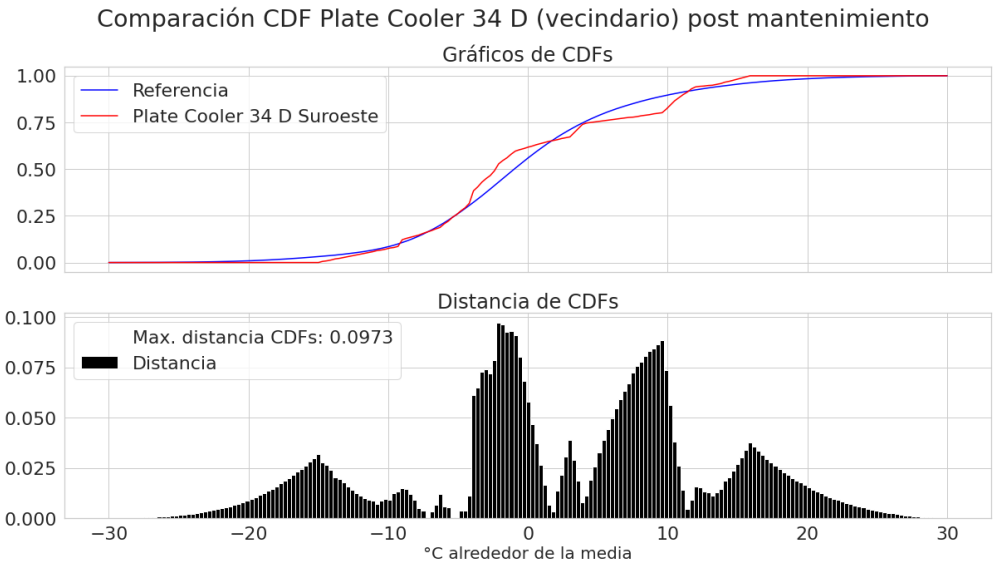


Figura 5-50: *CDF* del *Plate Cooler* 34 D después del mantenimiento

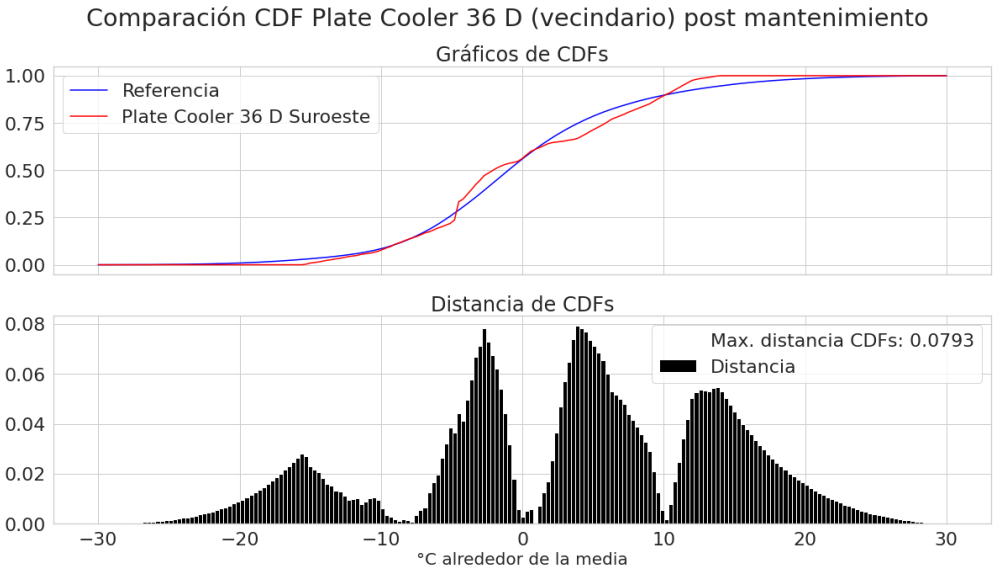


Figura 5-51: *CDF* del *Plate Cooler* 36 D después del mantenimiento

Se observa en estos casos que la máxima distancia entre las *CDF* de cada *Plate Cooler* y la referencia es apreciablemente diferente entre lo observado para el *Plate Cooler* 34 D (0.2940) y lo observado para los *Plate Coolers* 34 D y 36 D (0.0973, 0.0793 respectivamente).

Al repetir este tipo de análisis en diferentes termopares y ventanas de tiempo, se encontraron resultados similares, en los que no es claramente apreciable un valor umbral de diferencia de *CDF* para determinar la existencia de fallos en un termopar. Una posible causa identificada es el mayor rango de valores en condiciones de operación sin fallo que se manejan en el horno respecto al rango de valores en los que opera la planta utilizada por Mandal *et. al* [48]. La planta utilizada por Mandal *et al.* debe mantenerse entre 504 °C y 508 °C, con variaciones entre medidas consecutivas de algunas décimas de grado, mientras que el horno estudiando en el presente trabajo puede presentar valores desde los 40 °C hasta los 100 °C dentro de rangos de operación considerados normales y valores consecutivos registrados pueden variar, en algunos casos, en magnitudes de alrededor de 10 °C.

5.6. Selección de métricas

El uso de reglas heurísticas permitió detectar efectivamente la presencia de outliers y valores anclados en termopares individuales. Por lo que estos dos métodos se conservarán en su forma actual como método de detección y aislamiento de este tipo de fallos en los termopares del horno.

Respecto a la detección de fallos incipientes, con base en los resultados obtenidos en el presente capítulo, no es posible todavía la detección inequívoca de este tipo de fallos dado que no ha sido posible determinar métricas y valores umbral que caractericen efectivamente condiciones de normalidad o de fallo. Sin embargo, el análisis de correlación en vecindarios de termopares en ventanas de tiempo mostró una marcada relación entre valores altos en matrices de correlación y vecindarios de termopares libres de fallos.

Se decide conservar el análisis en ventanas de tiempo como herramienta para la segmentación de datos y el agrupamiento de termopares por vecindarios según tipo y nivel de panel como aproximación a la redundancia física en los termopares. A partir de esta estructura, y con base en los resultados del presente capítulo, se emplearán como parámetros los valores de la matriz de correlación del vecindario junto con las varianzas y medianas por ventana de cada panel para caracterizar cada ventana de tiempo de cada vecindario de termopares a analizar.

5.7. Establecimiento de línea base

A partir de los resultados obtenidos es posible usar reglas heurísticas para detectar y aislar fallos por outliers y valores anclados. Sin embargo, existen otros fallos que resultan de interés, como el fallo evidenciado en el *Plate Cooler* 35 D antes de la fecha 2015-04-05, y que no son fácilmente detectables todavía. Se decide entonces inspeccionar visualmente los datos en el tiempo de diferentes vecindarios (agrupados por tipo y nivel de panel) en ventanas de tiempo para identificar y etiquetar manualmente este tipo de fallos no detectables mediante reglas heurísticas y que se espera lograr detectar a través de técnicas basadas en datos.

5.7.1. Metodología

Se excluyen del análisis las combinaciones de termopares y ventanas de tiempo en las que se identifican fallos por valores anclados o presencia de outliers. Se agrupan todos los termopares del horno en vecindarios de entre 4 y 6 termopares y se grafica la totalidad de datos existentes para el vecindario en ventanas de ancho de dos semanas, cada combinación de vecindario de paneles y ventana de tiempo de dos semanas se denomina un segmento. Se inspeccionan visualmente los segmentos y se separan aquellos en que se identifica el tipo de fallo buscado, de los segmentos identificados .

Una vez identificados los segmentos con fallo, se procede a etiquetar manualmente los termopares, fechas y horas exactas en las que hay presencia o ausencia de fallos en cada segmento. La determinación del inicio y final del fallo se realiza mediante inspección visual de gráficos de líneas de tiempo de máximo 2 o 3 días en la abscisa. Para cada tipo y nivel de termopares se identificó una cantidad diferente de registros individuales con fallos.

5.7.2. Resultados y discusión

Se logra determinar visualmente la existencia de fallos como el mostrado en la figura **5-52**, donde se aprecia que las medidas obtenidas del *Waffle Cooler* 6 A presentan un fallo que hace que sus valores cambien bruscamente. Se puede determinar que este comportamiento corresponde con un fallo al compararlo con las demás señales mostradas en el segmento, correspondientes a termopares vecinos del *Waffle Cooler* 6 A.

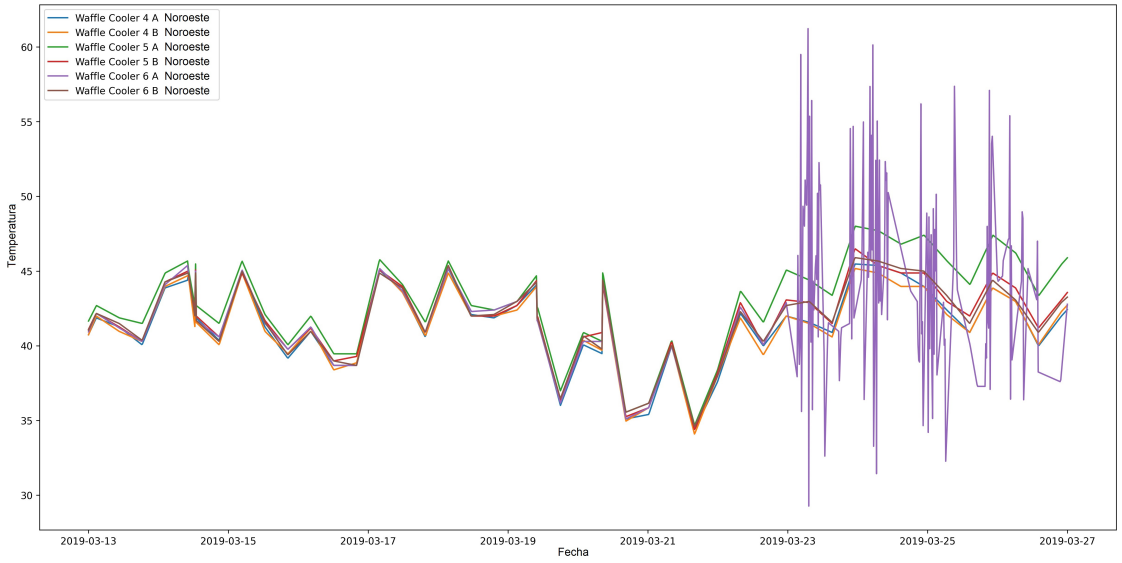


Figura 5-52: Ejemplo de segmento con fallo identificado

Del etiquetado manual de los termopares y marcas de tiempo con fallo se obtienen los resultados mostrados en la tabla **5-7**, donde se evidencia un desbalance en la cantidad de registros con fallo identificados en los diferentes tipos y niveles de termopares.

Termopar	Total Registros	Registros con fallo	Proporción en fallo
Plate Cooler A	21,500	11,732	0.5456
Plate Cooler B	648	264	0.4074
Plate Cooler C	10,512	5,424	0.5159
Plate Cooler D	5,136	2,592	0.5046
Waffle Cooler	7,056	4,128	0.5850
Total	44,852	24,140	0.5382

Tabla 5-7: Conteo de registros y fallos detectados con fallos mediante inspección visual

El ejercicio desarrollado no garantiza que se haya detectado visualmente el 100 % de los fallos en los termopares, pero permitió establecer una primera línea base de segmentos con fallos conocidos y algunos segmentos en los que se confirmó que no existen fallos. A partir de esta línea base se inicia el proceso de diseño e implementación de un modelo de *Machine Learning* que permita detectar esta clase de fallos identificados en la línea base.

6 Diseño de metodología de FDI

El estudio y selección de métricas para la detección de fallos en el horno ha permitido identificar, entre otras cosas, que el agrupamiento de termopares en vecindarios por tipo y nivel ofrece la posibilidad de sospechar de fallos a partir de los valores de la matriz de correlación. Sin embargo, hasta el momento no ha sido posible determinar un umbral o regla heurística para el etiquetado de un segmento como en fallo o libre de fallo.

El problema de detección de fallos en el horno presenta las siguientes características:

- Existen diversos modelos de *Machine Learning* aplicables al problema de clasificación.
- El horno se compone de 5 diferentes niveles de paneles de termopares (*Plate Coolers* A hasta D y *Waffle Coolers*) y se debe realizar el entrenamiento del modelo considerando el desbalance de registros en fallo de la tabla **5-7**.
- Se sabe que la segmentación en ventanas es de utilidad para facilitar la detección de fallos, pero no es posible establecer qué ancho de ventana es el más adecuado.

Cada característica mencionada da lugar a interrogantes para los que, con base en los resultados actuales del presente trabajo, no es posible aún determinar una respuesta. Es por esto que se aborda cada interrogante individualmente y se establece un diseño experimental que abarque las diferentes combinaciones de características del problema a resolver.

6.1. Selección de modelos de Machine Learning

Un sistema de detección de fallos puede verse como un problema de clasificación, donde se quiere determinar si una observación $\mathbf{x}$ pertenece a una de dos categorías: en fallo o sin fallo. Conocer el tipo de problema a resolver nos permite primeramente enfocarnos el los modelos de clasificadores, descartando bajo este criterio a los modelos regresores. La tabla **6-1** muestra la clasificación de diversos modelos de *Machine Learning* entre clasificadores y regresores.

Clasificador	Regresor
Árboles de decisión	Regresor lineal
Naive Bayes	Regresor logístico
Random Forest	
Redes neuronales	
Support Vector Machines (SVM)	
XGBoost	

Tabla 6-1: Modelos de Machine Learning de aprendizaje supervisado

Dos de los retos asociados al problema de clasificación actual son que el tipo de fallo que se desea detectar mediante el modelo de *Machine Learning* puede suceder en cualquiera de los termopares del segmento que se esté evaluando, y que el tipo de fallo de interés no tiene un único comportamiento (como los outliers o los valores anclados) sino que es una desviación respecto al comportamiento de los demás miembros del vecindario. Estas dos características hacen que sea de gran importancia la robustez del modelo frente a cambios en las variables de entrada, y con base en la breve introducción mencionada en 1.7 se pueden descartar los árboles de decisión y los clasificadores *Naive Bayes*.

6.1.1. Bosque Aleatorio (Random Forest)

Uno de los inconvenientes que se plantearon de los modelos de árboles de decisión es su tendencia al *overfitting*, es decir, a memorizar en lugar de aprender. Una de las primeras soluciones propuestas para reducir esta clase de efectos fue propuesta por Leo Breiman en 1996 y se conoce como *bagging* [50]. Esta técnica se basa en la selección aleatoria de muestras para el entrenamiento de varios modelos y la posterior combinación (agregación) de los resultados de cada modelo individual para obtener un resultado más acertado que empleando un solo modelo entrenado con la totalidad de los datos de entrenamiento. La selección de muestras para entrenar cada modelo se hace aleatoriamente y con reemplazo, es decir que una misma muestra puede aparecer en varios modelos.

Los modelos *Random Forest* utilizan una variante de *bagging* en la que cada árbol del bosque aleatorio es entrenado con un grupo de *features* seleccionados aleatoriamente y con reemplazo del grupo completo de *features* en el *dataset de entrenamiento*. Esto hace que el bosque aleatorio sea un conjunto de árboles no relacionados que, en conjunto, ofrecen una predicción más acertada que entrenar un único árbol con todos los *features*. [51]. La figura **6-1** muestra las estructuras de un modelo de árbol de decisión y un modelo de bosque aleatorio.

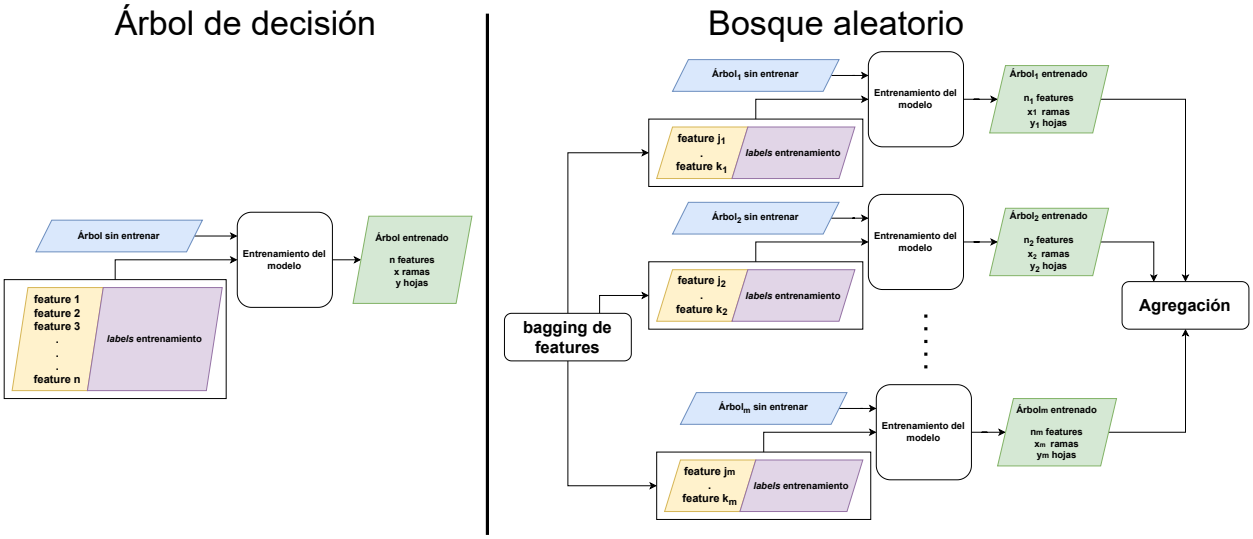


Figura 6-1: Estructuras de modelos de Árbol de decisión (izq.) y Bosque aleatorio (der.)

El modelo *Random Forest* ha sido empleado en tareas de clasificación como:

- Clasificación de imágenes de enfermedades en hojas de arroz. [52]
- Detección de fallos por arco eléctrico en redes de baja tensión a partir de análisis de señales en tiempo y frecuencia. [53]
- Detección de fallos en rotores mediante la clasificación de transformaciones de reducción de dimensionalidad a señales de vibración y presión. [54]
- Detección de fallos en sistemas de generación eólica mediante clasificación de parámetros extraídos a partir de reducción de dimensionalidad. [55]

León-Medina *et al.* [56] emplearon el modelo *Random Forest* para predecir la temperatura de hasta 12 puntos de la pared media de uno de los hornos mediante el uso de regresores multiobjetivo. El regresor emplea un conjunto de 52 variables de la operación del horno, compuesto por las medidas eléctricas de los electrodos (tensión, intensidad de corriente, potencia e impedancia), posición del electrodo, tap del transformador de alimentación, masa de calcina que ingresa al horno y composición química de la calcina. Los modelos desarrollados predijeron acertadamente la temperatura de hasta 12 puntos del horno en simultáneo, con un error medio absoluto ($AMAE$) máximo por debajo de los 3 °C.

6.1.2. Redes Neuronales (Neural Networks)

Las redes neuronales, o *NN* por su sigla en inglés, son un modelo de *Machine Learning* que se compone de nodos individuales denominados neuronas las cuales se conectan de manera similar a como se conectan las neuronas en el cerebro humano y forman una red que se distribuye en capas, similar al ejemplo que se muestra en la figura **6-2**.

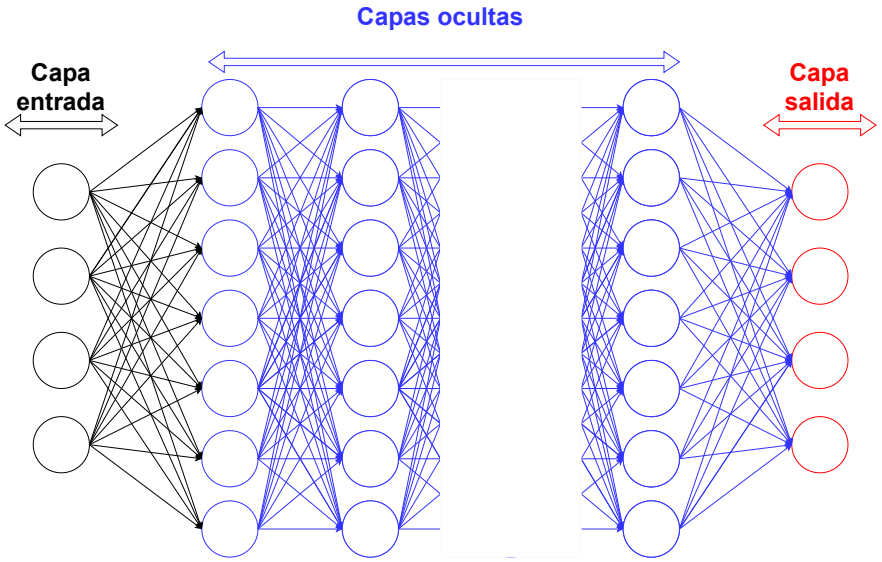


Figura 6-2: Ejemplo de estructura de una red neuronal

Cada nodo o neurona es un regresor individual que puede describirse con un modelo como el de la ecuación (11), donde se describe la función que caracteriza a una neurona **n** en la capa **c** en función de los resultados de las **m** neuronas de la capa $(\mathbf{c}-\mathbf{1})$ inmediatamente anterior y los pesos **w** que se les asignan para atenuar o acentuar la incidencia de cada uno.

$$f_{n,c}(x) = \sum_{i=1}^{m}(x_{i,c-1})(w_{i,c}) + sesgo = y_{n,c} \qquad (11)$$

El valor calculado se emplea como variable independiente del segundo componente de la neurona, la función de activación. Esta función, descrita en la ecuación (12), toma el valor calculado de la neurona y determina si se debe enviar la señal de activación a la neurona inmediatamente siguiente. La figura **6-3** muestra un esquema de una neurona con sus funciones de suma y activación.

$$f(y_{n,c}) = \begin{cases} 1, & \text{si } y_{n,c} \geq 0 \\ 0, & \text{si } y_{n,c} < 0 \end{cases} = z_{n,c} \qquad (12)$$

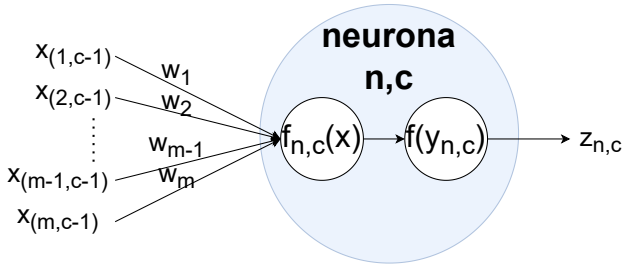


Figura 6-3: Esquema de una neurona

La construcción de una red neuronal requiere de la correcta combinación de cantidad de capas, cantidad de neuronas en cada capa (densidad), configuración de cada capa y entrenamiento de la red, para lo cual no existe un único método efectivo. Se pueden encontrar métodos en la literatura, como el propuesto por Anand y SaiDhiraj [57], donde se basan en características propias de un problema de modulación para establecer una arquitectura base para el entrenamiento de una red neuronal.

Algunos basados en redes neuronales incluyen los mencionados a continuación:

- Clasificación de consumo de energía eléctrica para identificar usuarios con necesidades no cubiertas. [58]
- Detección de fallos en juntas solapadas mediante análisis de señales de ultrasonido. [59]
- Detección de la dirección de llegada de señales electromagnéticas en arreglos de antenas. [60]
- Clasificación de redes de datos para establecimiento de clases de redes. [61]

En el marco de la colaboración entre la Universidad Nacional de Colombia y CMSA S.A. León-Medina *et al.* [62] emplearon una combinación de *gated recurrent unit (GRU)*, una capa densa y un modelo denso para predecir la temperatura de 16 termopares con base en en registros previos de variables como:

- Tensión, intensidad de corriente, potencia y posición de los electrodos.
- Masa de calcina enviada al horno.
- Datos históricos de temperatura.

Las predicciones obtenidas mediante este modelo presentaron un un error medio cuadrático (*RMSE*) de 1.19 °C respecto a los registros medidos en la planta.

Posteriormente, Godoy-Rojas *et al.* [63] hicieron uso de redes neuronales recurrentes para predecir la temperatura en diversos puntos de la pared media de uno de los hornos de CMSA empleando un conjunto de datos similar al empleado por León-Medina *et al.* Los autores lograron predecir la temperatura en diversos puntos de la pared media del horno con un error medio cuadrático (*RMSE*) máximo de 1.75 °C durante la etapa de validación. El estudio mostró adicionalmente que los mejores resultados de predicción se obtuvieron empleando ventanas de tiempo de 1 hora de duración.

6.1.3. Support Vector Machines (SVM)

Los modelos *SVM* fueron propuestos por Vapnik *et al.* [64] como un modelo de predicción robusto para clasificadores binarios a partir de la teoría de aprendizaje posteriormente conocida como teoría VC [65]. En este tipo de modelo de *Machine Learning*, dado un grupo de observaciones de entrenamiento $(x_i, y_i) \in R^d$, donde d es la dimensión del espacio de los datos de entrada, se busca definir un hiperplano[1] que divida el *dataset* de forma que los dos grupos de puntos, uno por cada etiqueta, queden en lados opuestos del hiperplano. Un hiperplano se define como se muestra en la ecuación (13), donde w es el vector normal al hiperplano

$$\mathbf{w}^T\mathbf{x} - b = 0 \qquad (13)$$

En caso de encontrarse al menos un hiperplano que separe los dos grupos de observaciones, se dice que el conjunto es linealmente separable y los datos quedan divididos como se muestra en la figura **6-4**. De todos los hiperplanos que puedan separar los dos grupos de datos, aquel que tenga la mayor distancia desde el hiperplano al punto más cercano de todo el conjunto se conoce como hiperespacio de separación óptima (*OSH*). La distancia libre que hay entre un hiperplano y el punto más cercano se conoce como margen.

En caso de que los datos no sean separables linealmente, se acude a la función de pérdidas denominada *hinge loss* [66] mostrada en la ecuación (14).

$$max(0, 1 - y_i(\mathbf{w}^T\mathbf{x} - b)) \qquad (14)$$

Donde y_i es la función objetivo de maximización de distancia del punto más cercano al hiperplano. El objetivo de la optimización es minimizar la expresión de la ecuación 15.

$$\lambda\|w\|^2 + \left[\frac{1}{n}max(0, 1 - y_i(\mathbf{w}^T\mathbf{x} - b))\right] \qquad (15)$$

[1] Espacio en la dimensión R^{d-1} derivado del espacio original al que pertenecen las muestras de entrenamiento.

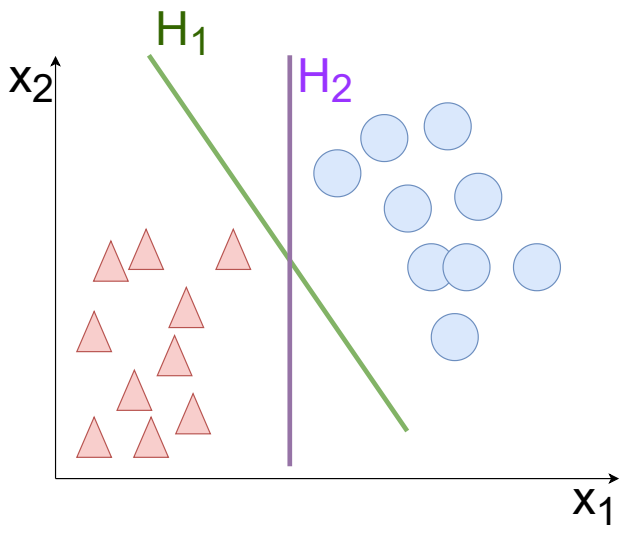


Figura 6-4: Esquema de división de dos grupos de observaciones mediante dos hiperplanos H_1 y H_2

Donde el parámetro $\lambda > 0$ determina el balance entre incrementar el tamaño de la margen libre y asegurar que todas las observaciones quedan del lado correcto de la margen. La forma como se resuelve el problema de optimización se conoce como el kernel del modelo.

Los modelos de *SVM* han sido empleados en desarrollos tales como:

- Detección de fallos en transformadores de potencia. [67]
- Predicción de volúmenes de carga para flotas de transporte. [68]
- Detección de anomalías en calidad de potencia mediante análisis de datos en frecuencia. [69]
- Análisis de emociones en redes sociales para predicción del precio de algunas acciones. [70]

6.1.4. Extreme Gradient Boosting (XGboost)

El modelo *XGBoost* [71] se desarrolló como una variante del algoritmo de *Gradient Tree Boosting* en el que se emplean árboles anidados para refinar la predicción del modelo a medida que se escala por las ramas del conjunto de árboles. Este proceso de mejorar la predicción del árbol m con base en los resultados del árbol inmediatamente anterior puede ser atenuado o acentuado modificando el peso ω_q de cada factor de entrada. Se define la salida de modelo como se muestra en la ecuación 16.

$$\hat{y}(x) = \sum_{k=1}^{n} (\omega_q)(x_i) \qquad (16)$$

El modelo *Gradient Tree Boosting* busca reducir la función de pérdida $\mathcal{L}$, descrita en la ecuación 17, a medida que se generan y entrenan nuevos árboles en el modelo.

$$\mathcal{L} = \sum_{i=1}^{n} r(\hat{y}_i, y_i) + \sum_{k=1}^{K} \Omega(f_k) \qquad (17)$$

Donde cada $r(\hat{y}_i, y_i)$ es la función de pérdida que mide el error entre los valores predichos y objetivos durante el entrenamiento del modelo empleando un número n de observaciones, y los $\Omega(f_k)$ son funciones de regularización para evitar tanto *overfitting* como *underfitting* en el modelo.

XGBoost incorpora tanto el algoritmo de optimización *Greedy* exacto de J. H. Friedman [72], como un algoritmo aproximado que se encarga de seleccionar las mejores formas de ramificar los árboles en cada nueva iteración. La función de pérdidas combinada en cada nueva ramificación viene dada como se muestra en la ecuación 18.

$$L_{split} = \frac{1}{2}\left[\frac{\left(\sum_{i\epsilon I_L} g_i\right)^2}{\sum_{i\epsilon I_L} h_i + \lambda} + \frac{\left(\sum_{i\epsilon I_R} g_i\right)^2}{\sum_{i\epsilon I_R} h_i + \lambda} - \frac{\left(\sum_{i\epsilon I} g_i\right)^2}{\sum_{i\epsilon I} h_i + \lambda}\right] - \gamma \qquad (18)$$

Donde cada g_i es el gradiente estadíastico de primer orden de la función de pérdidas, h_i es el de segundo orden, los factores de regularización λ y γ previenen tanto *overfitting* como *underfitting* en el modelo.

XGBoost inicia a partir de las hojas del árbol inicial y empieza a añadir nuevas ramas en cada iteración (ramificación). I_L e I_R son instancias de los nodos izquierdo y derecho, respectivamente, tras cada ramificación que se suman para obtener el factor I.

Debido al costo computacional que implica calcular todas las posibles combinaciones de las ramificaciones, el algoritmo *Greedy* aproximado de *XGBoost* filtra combinaciones no deseables con base en percentiles de distribución de *features* y selecciona la ramificación más apropiada.

Algunos proyectos desarrollados empleando *XGBoost incluyen*:

- Selección de estrategias en el mercado bursátil con base en predicciones del coeficiente de información. [73]
- Detección de casos de arritmia cardiaca con base en lecturas de Electrocardiograma. [74]

- Predicción de porosidad en minerales para en exploración petrolera. [75]
- Detección de fallos en generadores eólicos a partir de parámetros de funcionamiento ponderados. [76]
- Detección de fallos en sistemas de generación fotovoltaica con base en medidas de funcionamiento en línea. [77]

6.2. Selección de arquitectura y modo de entrenamiento

Cuando se habla de arquitectura se hace referencia al criterio que se empleará para agrupar termopares para entrenar modelos de *Machine Learning*. En este punto juega un papel importante el desbalance presentado en la tabla **5-7** para los *Plate Coolers* B, donde la cantidad de fallos identificados es significativamente menor respecto a los demás tipos de termopar, mientras que la cantidad de registros en fallo de los *Plate Coolers* A es notablemente mayor que el resto. Se plantean dos posibles modos de agrupamiento para el entrenamiento de modelos de *Machine Learning* con base en el agrupamiento de vecindarios:

- **Arquitectura por tipo y nivel:** En esta opción se propone entrenar 5 modelos de *Machine Learning*, uno para cada tipo y nivel de termopar en el horno. Se intentará entrenar los modelos tanto con la metodología *k-folds* como con el proceso sin iterar para comparar resultados
- **Arquitectura de Modelo único:** Esta aproximación parte la siguiente hipótesis: el tipo de fallo de interés presenta comportamientos similares independientemente del tipo y nivel de panel, al punto en que se pueden combinar los datos de los 5 niveles para conformar un único dataset en el cual, en teoría, todos registros con y sin fallos tienen la misma relevancia y son independientes de su procedencia.

Además del modo de agrupamiento, es preciso establecer el tamaño de los vecindarios a utilizar. Se opta por utilizar vecindarios con un número impar de termopares para que, al tomar como referencia el panel central del vecindario, exista un número igual de paneles contra los cuales realizar comparaciones.

En cuanto al modo de entrenamiento, se sabe de antemano que algunos tipos y niveles de termopares tienen pocas muestras, por lo que se incluyen contempla realizar entrenamiento por *hold-out* y por *k-folds*.

6.3. Selección de ancho de ventana de tiempo

En el capítulo 5 se mostró la importancia del análisis en ventanas de tiempo, por lo que se considerará la variación de este parámetro para el diseño experimental. Considerando que es de interés la detección de fallos en intervalos cortos de tiempo (detectar los fallos transcurrido poco tiempo después de que ocurren), se incluirán ventanas de tiempo de máximo 1 día de duración.

6.4. Diseño experimental

A partir de los parámetros establecidos se propone realizar una variación de parámetros para las diferentes variables del proceso de entrenamiento de modelos de *Machine Learning*. Considerando que se plantea un total de 5 parámetros, como se muestra en la tabla **6-2**, se divide el proceso en dos ejercicios utilizando diseño centra compuesto para establecer los parámetros y valores en cada variación.

Párametro	**Valores**
Modelo ML	- *Random Forest (RF)* - Redes Neuronales *(NN)* - SVM - XGBoost
Modo de entrenamiento	- *hold-out* - *k-folds*
Arquitectura	- Por tipo y nivel - Modelo único
Paneles por vecindario	(3 , 5)
Ancho de ventana [horas]	(1, 3, 6, 12, 24)

Tabla 6-2: Parámetros y valores

En el primer ejercicio de variación de parámetros se fijarán los paneles por vecindario y el ancho de ventana para variar los restantes tres parámetros, como se muestra en la figura **6-5**.

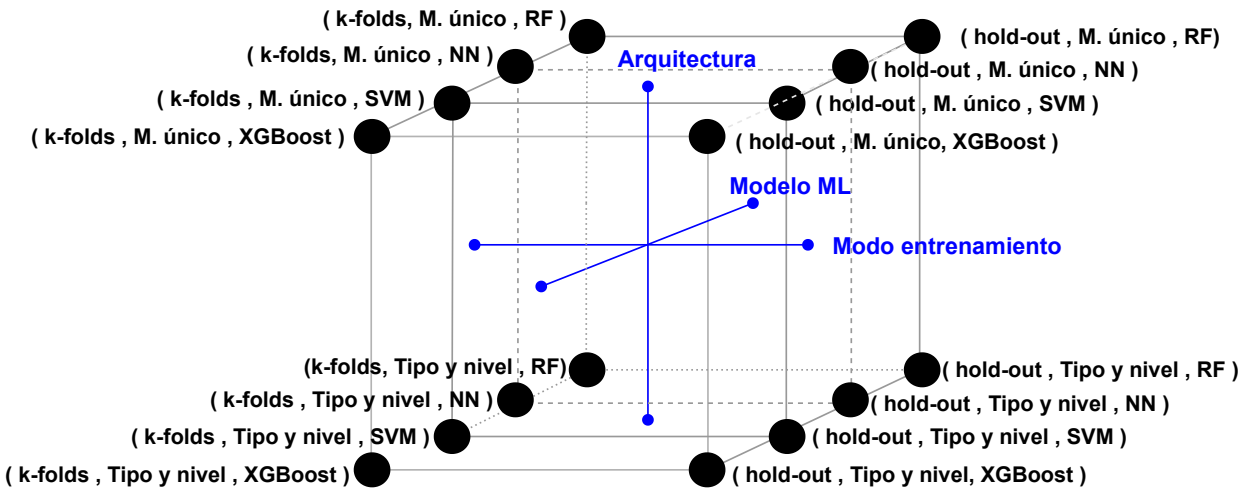


Figura 6-5: Diseño experimental fase 1

Con base en los resultados obtenidos de la fase 1 del diseño experimental, se fijarán la arquitectura, modelo de *Machine Learning* y modo de entrenamiento para realizar una nueva variación, esta vez en los dos parámetros previamente asumidos como fijos, de acuerdo con el esquema mostrado en la figura **6-6**. Los valores a conservar de estos primeros tres parámetros serán aquellos en los que se obtengan los mayores valores posibles de sensibilidad, especificidad y precisión.

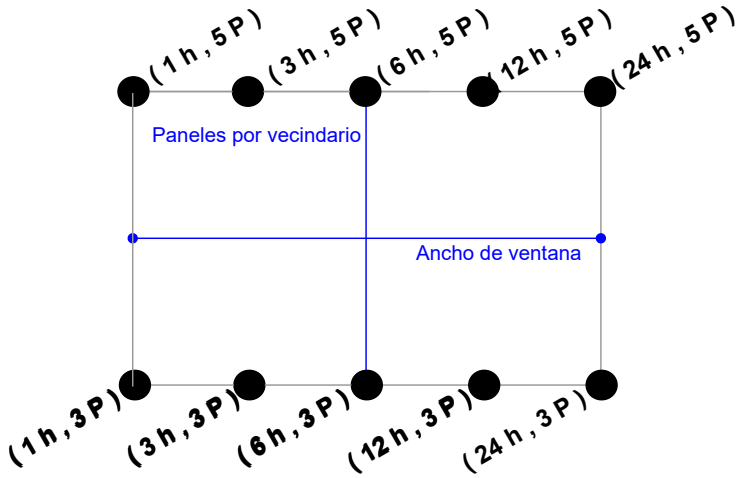


Figura 6-6: Diseño experimental fase 2

Los resultados de la ejecución de los diseños experimentales se muestran en el capítulo 7.

7 Entrenamiento de modelos

A partir de los diseños experimentales planteados en la sección 6.4 se desarrolla inicialmente el diseño experimental denominado **fase 1**, ilustrado en la figura **6-5**, y se comparan los resultados obtenidos en cada caso. Se mantienen fijos el ancho de ventana de 1 día y vecindarios de 5 paneles para limitar la cantidad de combinaciones de parámetros. Al finalizar el ejercicio se conserva la combinación de modo de entrenamiento, arquitectura y modelo de *ML* con la que se obtengan los mayores valores posibles de sensibilidad, especificidad y precisión durante la etapa de validación del modelo.

Se procede con la creación de los *datasets* de features y de *labels* para el entrenamiento de modelos.

7.1. Creación de datastets de features

A partir del establecimiento de una línea base presentada en la sección 5.7, tanto los registros etiquetados como çon falloçomo los etiquetados "libres de fallo"provienen de la observación y etiquetado manual de diferentes segmentos de mediciones, por lo que existen discontinuidades entre segmentos adyacentes, como se muestra en la figura **7-1** donde se presentan dos segmentos de datos de interés pertenecientes a *Plate Coolers* del nivel C.

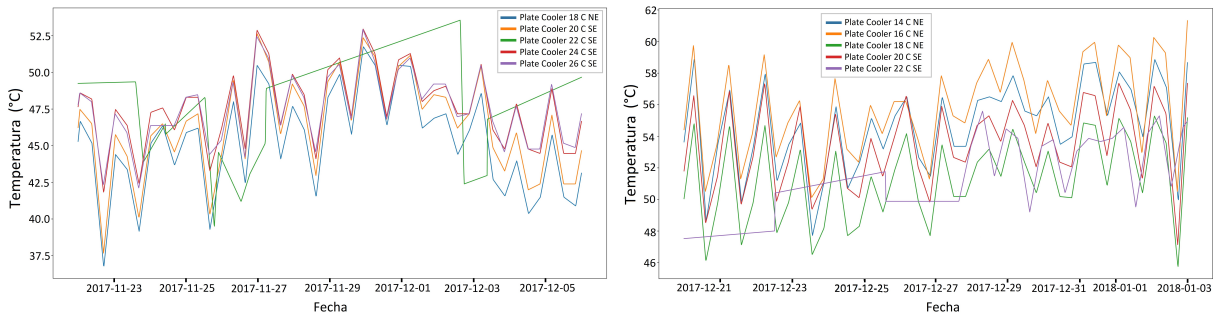


Figura 7-1: Segmentos de datos adyacentes de *Plate Coolers* del nivel C

Extraer una ventana de tiempo que incluya discontinuidades en los datos generaría datos que no corresponden con ningúna ventana de datos real del sistema. Por lo que se diseña un algoritmo para generar *features* en conjuntos de datos con discontinuidades.

7.1.1. Metodología

Tanto para la arquitectura denominada **por tipo y nivel** como para la arquitectura denominada **modelo único**, presentadas en la sección 6.2, se realiza un análisis en ventanas de tiempo para obtener el conjunto de *features* que caracteriza cada ventana siguiendo el diagrama mostrado en la figura **7-2**.

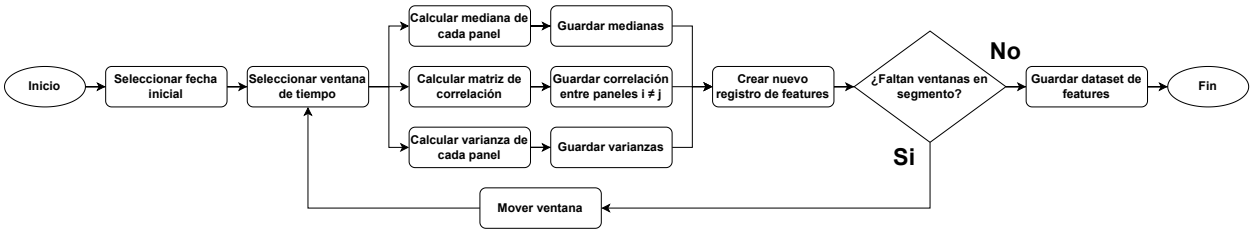


Figura 7-2: Diagrama de flujo de cálculo de *features*

La ventana móvil se desliza a razón de 1 registro (15 minutos) por iteración en un mismo segmento hasta que el borde final de la ventana coincide con el final del segmento, el siguiente segmento en el conjunto de datos se inicia haciendo coincidir la fecha inicial de la ventana con la fecha inicial del segmento, como se ilustra en la figura **7-3**.

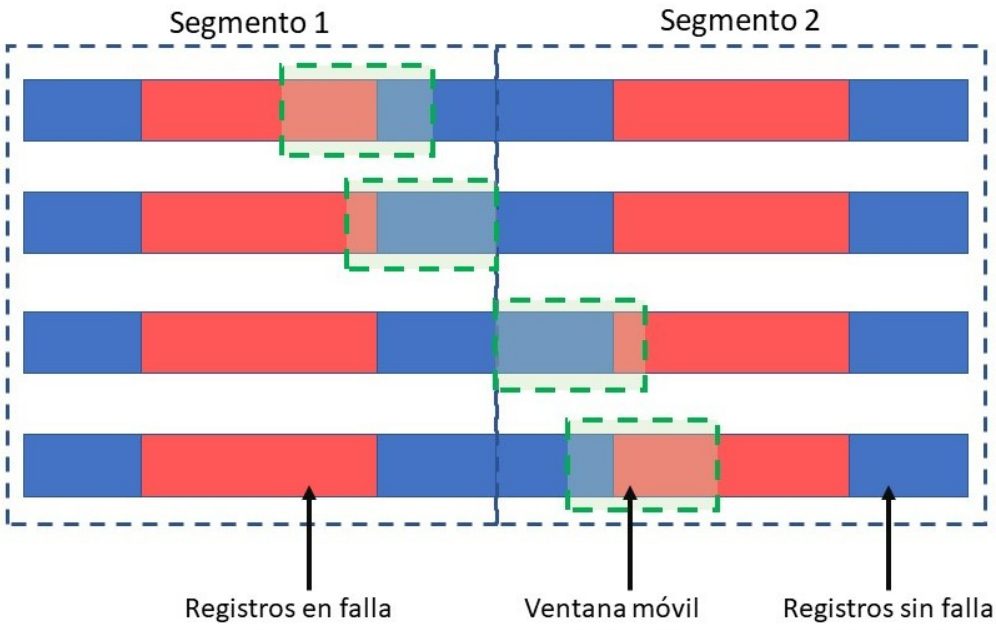


Figura 7-3: Esquema de movimiento de ventana deslizante

Ya que se selecciona una cantidad igual de paneles por vecindario en cada segmento, todos los datasets de *features* tienen la misma cantidad de variables y tipo de información, lo que hace posible unir datasets generados de segmentos adyacentes.

7.1.2. Resultados y discusión

Se obtienen 6 *datasets* de *features*, uno por cada fila de la tabla **5-7**. Cada *dataset* tiene 96 registros menos que el total de registros en cada fila, lo cual corresponde con el hecho que una ventana de 1 día de ancho tiene 96 registros espaciados cada 15 minutos y el último conjunto de features se calcula para el último conjunto de datos disponibles de 1 día de duración.

7.2. Creación de datastets de labels

Para la creación del dataset de labels se sigue el mismo principio de ventana deslizante utilizado para la creación del dataset de features en cuanto al manejo de la información proveniente de diferentes segmentos. En la figura **5-7** se aprecia que existen instantes en que la ventana móvil abarca tanto registros con fallos como registros sin fallos, con lo que se hace necesario establecer una regla de decisión sobre si la ventana se etiqueta como con fallo (1) o libre de fallos (0).

7.2.1. Metodología

Para definir si una ventana se etiqueta como con fallos se parte de la teoría de cálculo de tamaño de muestreo estadístico de una población, donde el tamaño de muestra **n** se puede calcular a partir del tamaño de la población mediante la expresión de la ecuación 19. [78]

$$n = \frac{Z^2 N p q}{N E^2 + Z^2 p q} \qquad (19)$$

Donde:

- **Nivel de confianza Z:** Se calcula mediante tablas de distribución de probabilidad, donde a un cierto nivel de confianza le corresponde un valor Z asociado. Un valor común en la bibliografía es de 95 % de confianza, correspondiente con un valor Z de 1.96
- **Tamaño de la población N:** Cantidad de registros que componen cada ventana
- **Error de muestreo E:** Margen de error aceptable entre el resultado de la muestra y el de la población, en la bibliografía se emplean valores menores o iguales a 10 %. Para el presente trabajo se emplea un error de muestreo de 10 % para permitir un menor tamaño muestral.
- **Probabilidades de eventos p y q:** Probabilidad de que sucedan los dos eventos a identificar en la muestra. Al ser los estados de fallo sin fallo dos eventos mutuamente excluyentes se tiene que $q = 1 - p$

7.2.2. Resultado y discusión

A partir de la ecuación 19 y los datos de la tabla **5-7** se calcula el tamaño muestral para cada tamaño de ventana. Los resultados se aproximan usando la función techo y muestran en la tabla **7-1**.

Tipo Termopar	**p(fallo)**	**horas en ventana (arr), cant. registros N (ab)**				
		1 hora	**3 horas**	**6 horas**	**12 horas**	**24 horas**
		4	**12**	**24**	**48**	**96**
Plate Coolers A	0.5456	4	11	20	32	48
Plate Coolers B	0.4074	4	11	20	32	48
Plate Coolers C	0.5159	4	11	20	32	48
Plate Coolers D	0.5046	4	11	20	33	49
Waffle Coolers	0.5850	4	11	20	32	48
Total	0.5382	4	11	20	32	48

Tabla 7-1: Tamaños de muestra por tipo de panel y ancho de ventana

Los valores obtenidos se interpretan, para efectos del etiquetado de ventanas en el presente trabajo, como la cantidad mínima de registros individuales con fallo que debe haber en una ventana para que esta sea etiquetada como una ventana con presencia de fallos en el *dataset* de *labels*.

7.3. Ejecución de experimento fase 1

Del proceso de validación de cada modelo de *Machine Learning* se obtiene una matriz de confusión como la mostrada en la figura **7-4**.
Empleando las ecuaciones 5 a 7 se obtienen los valores de sensibilidad, especificidad y precisión, respectivamente. Los resultados correspondientes a la matriz de confusión de la figura **7-4** se muestran a continuación:

- $Sensibilidad = \frac{\sum VP}{\sum VP + \sum FN} = \frac{7,616}{7,616 + 253} = 0{,}9678$
- $Especificidad = \frac{\sum VN}{\sum VN + \sum FP} = \frac{6,646}{6,646 + 255} = 0{,}9630$
- $Precisión = \frac{\sum VP + \sum VN}{\sum TP + \sum FN + \sum TN + \sum FP} = \frac{7,616 + 6,646}{7,616 + 253 + 6,646 + 255} = 0{,}9656$

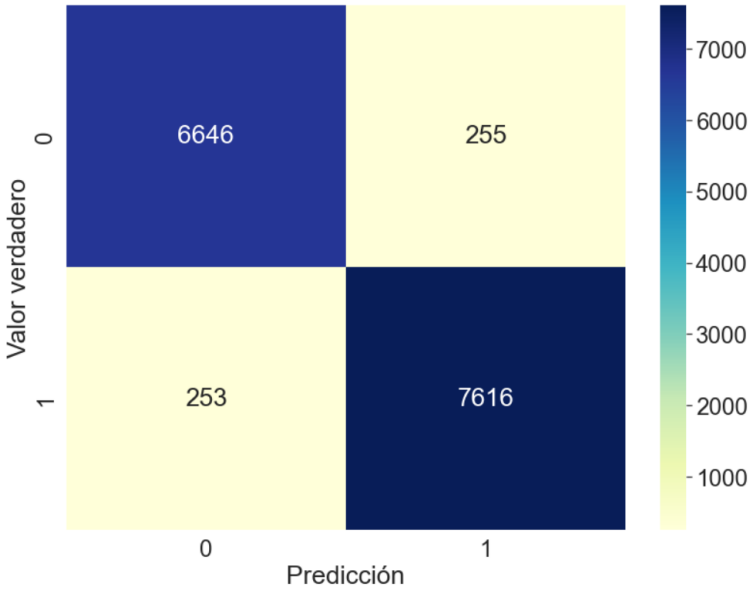


Figura 7-4: Ejemplo de matriz de confusión

Adicionalmente existen métricas que son complemento de la sensibilidad, la especificidad y la precisión, estas son: tasa de falsos negativos, tasa falsos positivos y error de clasificación, respectivamente. Se calculan mediante las expresiones de las ecuaciones 20 a 22.

$$Tasa\ FN\ = \frac{\sum FN}{\sum VP + \sum FN} = 1\ -\ Sensibilidad \qquad (\ 20\ )$$

$$Tasa\ FP\ = \frac{\sum FP}{\sum VN + \sum FP} = 1\ -\ Especificidad \qquad (\ 21\ )$$

$$Error\ clasificación = \frac{\sum FN + \sum FP}{\sum VN + \sum FP} = 1\ -\ Precisión \qquad (\ 22\ )$$

Empleando estas dos métricas derivadas se puede crear una versión de la matriz de confusión que agrupa las cuatro métricas, permite sintetizar el resultado de la validación y además emplear un eje de referencia normalizado respecto a la cantidad de observaciones disponibles en el *dataset*. La forma modificada de la matriz de confusión se lee siguiendo el esquema de la figura **7-5**.

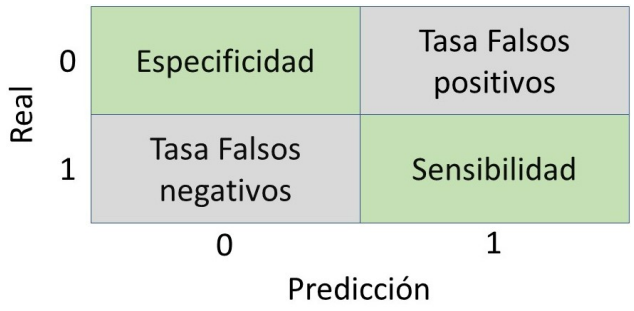


Figura 7-5: Esquema de matriz de confusión normalizada

En la figura **7-6** se presenta la matriz de confusión normalizada para el ejemplo presentado en la figura **7-4**

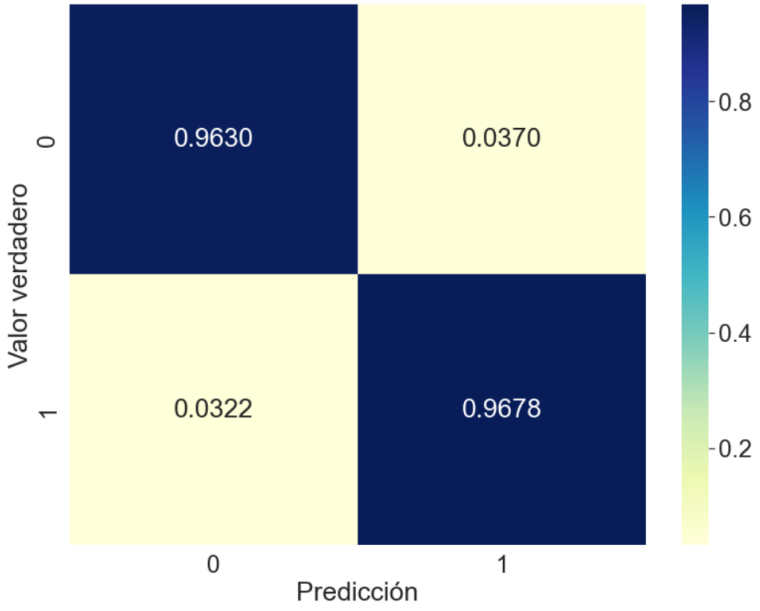


Figura 7-6: Ejemplo de matriz de confusión normalizada

La matriz de confusión refleja el resultado final de la validación del modelo. Una forma de obtener información sobre si nuestro modelo tiene un buen ajuste a los datos de entrenamiento es a través del estudio de las curvas de aprendizaje.

7.3.0.1. Curvas de aprendizaje

La curva de aprendizaje de un modelo de *Machine Learning* relaciona gráficamente una métrica de desempeño de un clasificador (eje de las ordenadas) en función de un parámetro de entrenamiento (eje de la abscisa) [79], tanto el parámetro de entrenamiento que se varía como la métrica de desempeño se seleccionan en función del modelo que se evalúa. En la figura **7-7** se presentan dos curvas de aprendizaje de un modelo *XGboost* durante las etapas de entrenamiento y validación.

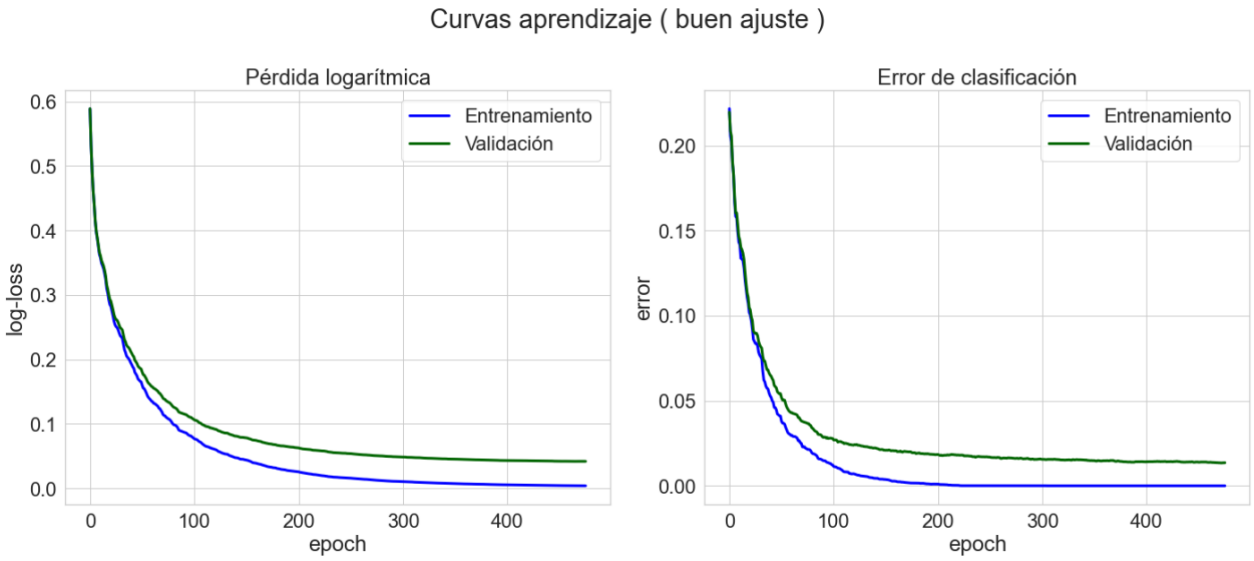


Figura 7-7: Ejemplo de curvas de aprendizaje con buen ajuste

El parámetro del modelo que se selecciona y se modifica para buscar un mejor ajuste se denomina **hiperparámetro**. En el ejemplo de la figura **7-7** el hiperparámetro mostrado en el eje de la abscisa es la cantidad de *epochs*[1] de un modelo *XGBoost* y se presentan dos métricas de desempeño en los ejes de la ordenada, la pérdida logarítmica y el error de clasificación. Estudiar la función de pérdidas y el error de clasificación de un modelo puede ayudar a determinar el tipo de ajuste obtenido en función de un hiperparámetro.

Las curvas mostradas en la figura **7-7** corresponden a un modelo bien ajustado, donde las pérdidas logarítmicas y el error de clasificación decaen de forma similar a medida que se realizan más *epochs* hasta alcanzar un punto estable tanto en entrenamiento como en validación. Se observa que el error de clasificación deja de decaer y llega a un valor mínimo, a partir de ese punto pueden presentarse dos tipos de ajuste a medida que se sigue incrementando el valor del hiperparámetro:

[1]Se denomina *epoch* al proceso de hacer pasar la totalidad de los datos de entrenamiento a través de un modelo.

- **Ajuste invariante:** Tanto la pérdida como el error de clasificación se mantienen en valores mínimos alcanzados en un valor menor del hiperparámetro, como se presenta en la figura **7-8**. Implica que el incremento posterior del hiperparámetro no genera ninguna diferencia en el desempeño y que puede obtenerse un buen modelo con un valor menor del hiperparámetro manteniendo el mismo error de generalización [2].

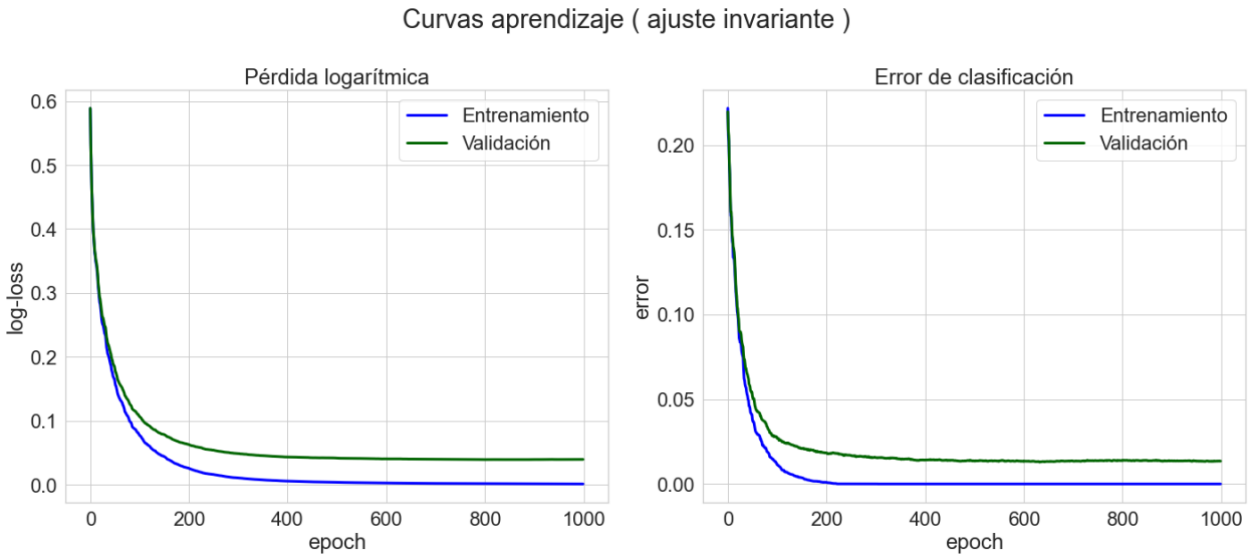


Figura 7-8: Ejemplo de curvas de aprendizaje con ajuste invariante

- **Sobre ajuste:** En este escenario se pueden encontrar que el error de clasificación cae a cero y el error de generalización crece a medida que crece el hiperparámetro, como se muestra en la figura **7-9**, esto implica que el modelo está empezando a memorizar el *dataset* de entrenamiento y pierde su capacidad de generalizar. Debido a que el sobre ajuste puede deberse a diversas causas, la estrategia para mejorar el desempeño del modelo depende de la causa.

[2]Diferencia entre los valores del error de entrenamiento y validación.

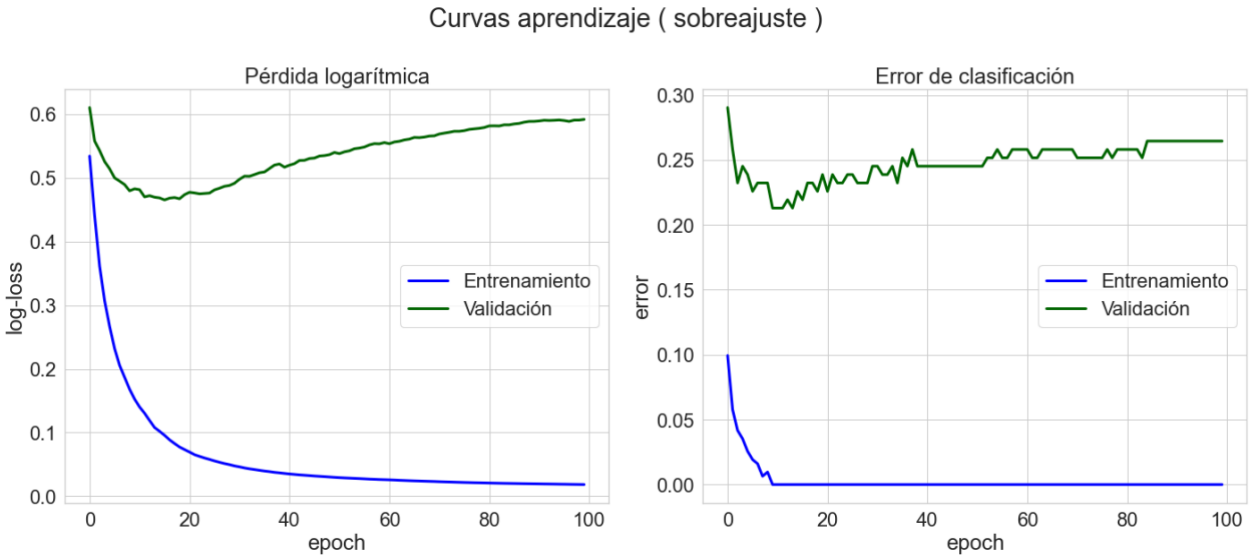


Figura 7-9: Ejemplo de curvas de aprendizaje con sobreajuste

7.3.1. Metodología

De la ejecución de cada punto del diseño experimental propuesto en la figura **6-5** se obtienen las matrices de confusión normalizadas de cada punto, se calcula la precisión del proceso de validación siguiendo la ecuación 7 y se selecciona la combinación de presente los valores más altos de sensibilidad, especificidad y precisión. Debido a que la arquitectura **Por tipo y nivel** implica crear 6 modelos en cada punto del diseño experimental, se mostrarán separados los resultados de cada arquitectura.

Se realiza análisis de curvas de aprendizaje en algunos puntos de interés del diseño experimental en los que los resultados dan indicios de la existencia de sobre ajuste.

7.3.2. Resultados y discusión

En la tabla **7-2** se presenta una comparación de los resultados de la validación del entrenamiento de los diferentes modelos de *Machine Learning* usando la arquitectura de **modelo único**. Se resaltan en color verde los modelos que presentaron los valores más altos de sensibilidad, especificidad y precisión.

Modelo ML	Modo de entrenamiento					
	k-folds			hold-out		
	sens.	espec.	pres	sens.	espec.	pres
Random Forest	0.9676	0.9617	0.9649	0.9677	0.9629	0.9655
Neural network	0.8867	0.8671	0.8876	0.8583	0.8778	0.8674
SVM	0.4533	0.9213	0.6695	0.4612	0.9251	0.6779
XGBoost	0.9659	0.9579	0.9622	0.9671	0.9640	0.9657

Tabla 7-2: Resultados de entrenamiento de modelos para arquitectura de **modelo único**

Se observa que los resultados para los modelos de *Random Forest* y *XGBoost* son muy similares, diferenciándose por unas pocas cifras decimales. Siguiendo el criterio de decisión establecido, se selecciona el modelo *XGBoost* ya que presenta los mejores resultados en 2 de las 3 métricas, siendo superado por el modelo *Random Forest* solamente en cuanto a sensibilidad, por una diferencia de 0.01 %.

Como segunda observación, se aprecia que los resultados obtenidos entrenando mediante *k-folds* no presentan diferencias marcadas respecto a los entrenamientos usando *hold-out*, esto puede deberse a que el *dataset* de entrenamiento y validación se generó de forma tal que fuera lo más balanceado posible y cuidando de que tanto los registros en fallo como los registros libres de fallo presenten características similares, lo que ocasiona que en cada *fold* el modelo se entrene con datos muy similares.

En tercer lugar se observa que el modelo de *Support Vector Machine (SVM)* presenta una sensibilidad por debajo del 50 % sin importar el modo de entrenamiento, lo cual resulta muy inferior a los demás modelos evaluados.

Durante las primeras etapas de la ejecución del diseño experimental para la arquitectura **por tipo y nivel** se evidenciaron resultados como los mostrados en la tabla **7-3**. Se resaltan las filas donde se quiere llamar especialmente la atención del lector.

Modelo ML	Tipo y nivel	Modo de entrenamiento					
		hold-out			k-folds		
		sens.	espec.	pres	sens.	espec.	pres
Random Forest	Plate Cooler A	0.9396	0.9301	0.9353	0.9360	0.9225	0.9299
	Plate Cooler B	0.9677	1.0000	0.9891	0.9861	1.0000	0.9946
	Plate Cooler C	0.9983	0.9970	0.9977	0.9987	0.9982	0.9985
	Plate Cooler D	0.9976	1.0000	0.9988	0.9988	0.9992	0.9990
	Waffle Cooler	0.9993	0.9968	0.9983	0.9993	0.9965	0.9981
XGBoost	Plate Cooler A	0.9411	0.9301	0.9362	0.9380	0.9227	0.9311
	Plate Cooler B	0.9677	1.0000	0.9891	0.9861	0.9940	0.9909
	Plate Cooler C	0.9989	1.0000	0.9994	0.9993	0.9988	0.9990
	Plate Cooler D	0.9917	0.9976	0.9946	0.9961	0.9984	0.9972
	Waffle Cooler	0.9985	0.9968	0.9978	0.9985	0.9955	0.9973

Tabla 7-3: Resultados preliminares entrenamiento por tipo y nivel en arquitectura **por tipo y nivel**

Se observan valores de especificidad de 100 % y valores precisión arriba del 99 %, lo cual suele ser un indicio de sobre ajuste en el modelo. Se estudian las curvas de aprendizaje de los modelos resaltados en la tabla **7-3**.

Las figuras **7-10** y **7-11** muestran las curvas de aprendizaje para el modelo *Random Forest* de los *Plate Coolers* tipo B y D, respectivamente. Se observa en ambos casos que el error de clasificación se vuelve 0 desde el inicio del eje de las abscisas para los datos de entrenamiento, mientras que el valor es variable con los datos de entrenamiento, indicando que el modelo sufre de sobre ajuste.

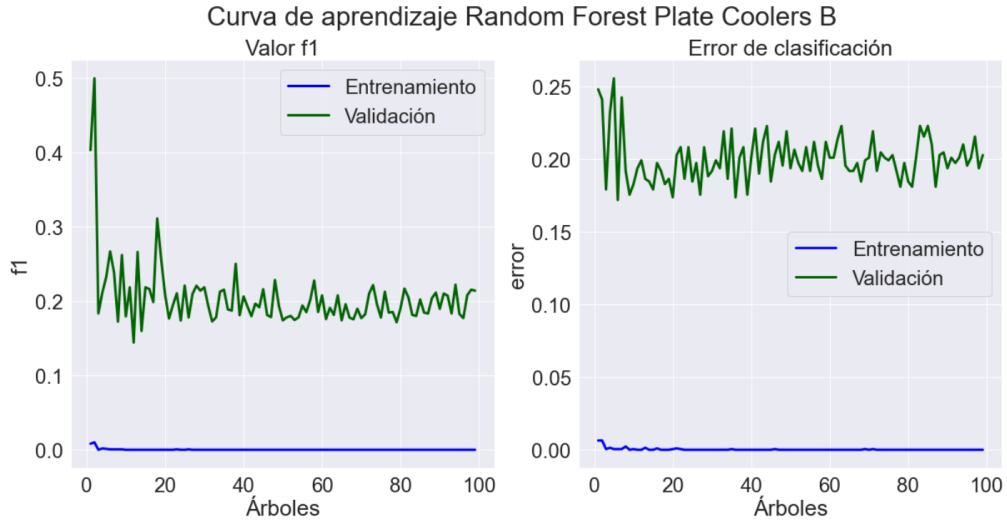


Figura 7-10: Curvas de aprendizaje *hold-out* modelo *Random Forest* con *Plate Coolers B*

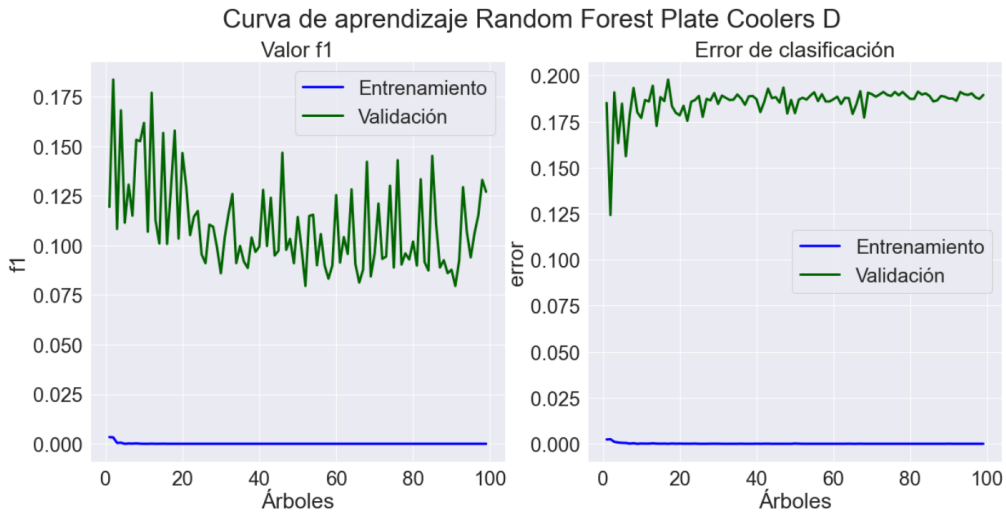


Figura 7-11: Curvas de aprendizaje *hold-out* modelo *Random Forest* con *Plate Coolers D*

En las figuras **7-12** y **7-13** se observan las curvas de aprendizaje para el modelo *XGBoost* de los *Plate Coolers* tipo B y C. En el caso de los *Plate Coolers B* se observa nuevamente que el error de clasificación en entrenamiento siempre es 0, correspondiente con la memorización de los datos de entrenamiento, mientras que el error de clasificación presenta dos únicos valores y ninguna otra incidencia del hiperparámetro en el modelo. En el caso de los *Plate Coolers C* la evidencia radica en la ausencia de error de generalización, lo que indica que el conjunto de datos es demasiado pequeño y el modelo aprende los datos en lugar de establecer una generalización.



Figura 7-12: Curvas de aprendizaje *hold-out* modelo *XGBoost* con *Plate Coolers B*

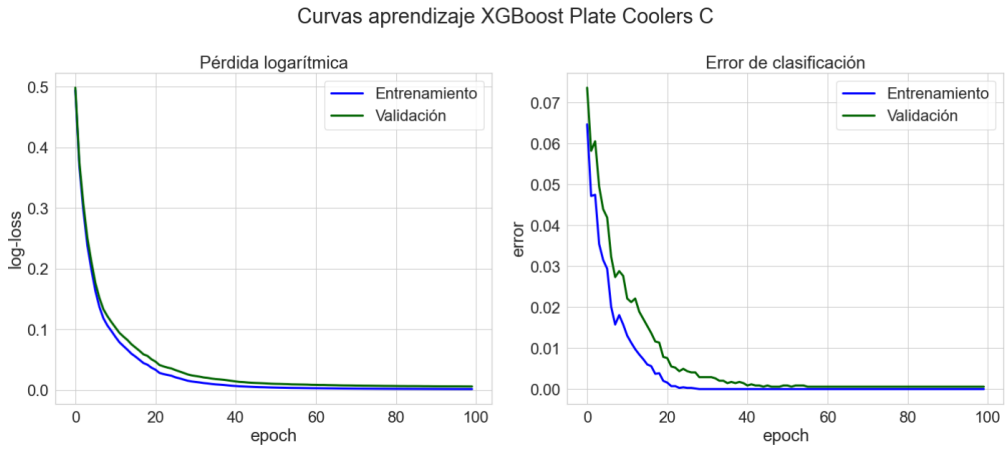


Figura 7-13: Curvas de aprendizaje *hold-out* modelo *XGBoost* con *Plate Coolers C*

Al encontrarse este comportamiento en los modelos con los mejores desempeños registrados en la arquitectura **modelo único** se decide descartar la arquitectura **por tipo y nivel**, de forma que el resultado de la **fase 1** es emplear un modelo *XGBoost* entrenado usando el método *hold-out*.

7.4. Ejecución de experimento fase 2

Los modelos entrenados durante la **fase 1** conservan los hiperparámetros por defecto para reducir la cantidad de ejes del diseño experimental de esta fase. Durante el desarrollo de la **fase 2** se realiza una variación del hiperparámetro de cantidad de *epochs* del modelo *XGBoost* para buscar un mejor desempeño en cada punto del modelo experimental descrito en la figura **6-6**.

Para realizar la variación del hiperparámetro se emplea un método denominado *early stopping* [80]. En este algoritmo se monitorea la evolución del desempeño del modelo en función de si las pérdidas disminuyen en cada nueva iteración, deteniendo la variación del hiperparámetro si el modelo no mejora su desempeño luego de un determinado número de variaciones del hiperparámetro. Los resultados se muestran en las figuras **7-14** y **7-15**.

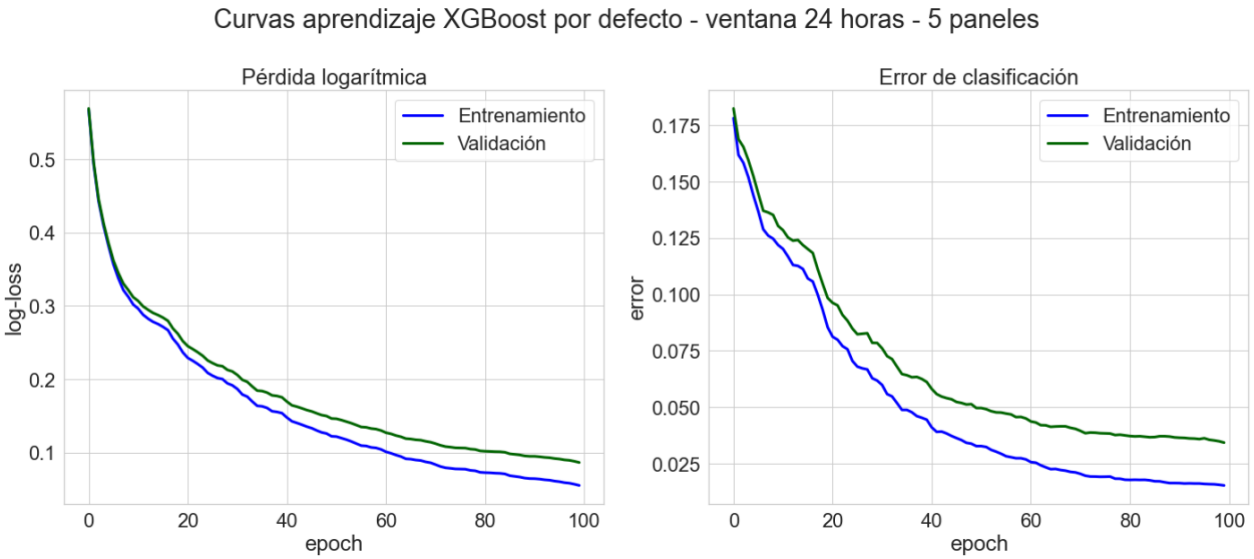


Figura 7-14: Curvas de aprendizaje modelo *XGBoost* ventana de 24 horas, 5 paneles, sin *early stopping*

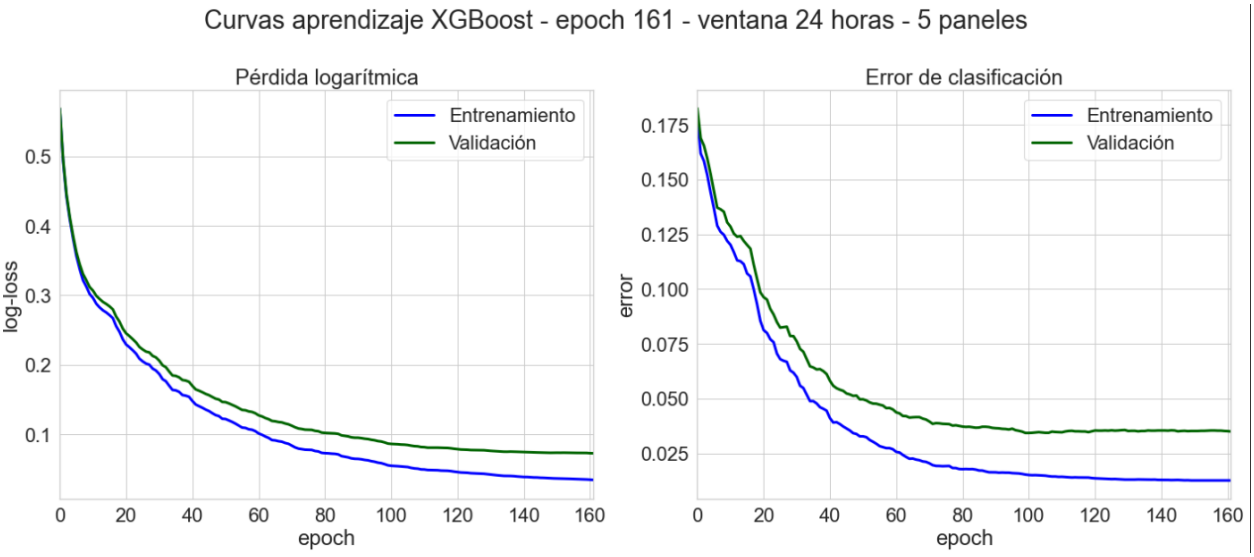


Figura 7-15: Curvas de aprendizaje modelo *XGBoost* ventana de 24 horas, 5 paneles, con *early stopping*

7.4.1. Metodología

Se entrenan modelos *XGBoost* con los parámetros indicados en la figura **6-6**, variando la cantidad de paneles por vecindario y la cantidad de horas por ventana de tiempo, se realiza una variación del hiperparámetro *epoch* empleando la metodología *early stopping* para buscar un mejor desempeño del modelo. Al finalizar el ejercicio se conserva el tamaño de vecindario y ancho de ventana con el que se hayan obtenido los mayores sensibilidad, especificidad y precisión.

7.4.2. Resultados y discusión

Los resultados del ejercicio se sintetizan en la tabla **7-4**. Se resaltan dos casos especiales, en los que se identificó en las curvas de aprendizaje que el modelo tenía tendencia al sobreajuste, como se muestra en las figuras **7-16** y **7-17**. Los nombres de las columnas se muestran abreviados.

Paneles en vecindario	**Horas en ventana**	**Hiperparametro**						**epochs finales**
		Por defecto			**Modificado**			
		Sens.	**Espec.**	**Pres.**	**Sens.**	**Espec.**	**Pres.**	
5	**24**	0.9671	0.9640	0.9657	0.9674	0.9623	0.9650	161
	12	0.9750	0.9908	0.9825	0.9975	0.9978	0.9975	499
	6	0.9506	0.9725	0.9611	0.9506	0.9725	0.9611	647
	3	0.9528	0.9662	0.9591	0.9860	0.9868	0.9864	551
	1	0.9723	0.9798	0.9758	0.9856	0.9876	0.9865	471
3	**24**	0.9402	0.9426	0.9423	0.9546	0.9529	0.9538	226
	12	0.8443	0.8882	0.8652	0.8665	0.9002	0.8825	281
	6	0.8073	0.8933	0.8485	0.8456	0.8907	0.8672	298
	3	0.8799	0.9141	0.8959	0.9063	0.9141	0.9100	298
	1	0.9346	0.9400	0.9371	0.9462	0.9433	0.9449	299

Tabla 7-4: Resultados diseno experimental fase 2

En las figuras **7-16** y **7-17**, mostradas a continuación, se observa que el modelo por defecto muestra una tendencia en el comportamiento de las pérdidas y el error de clasificación a adquirir mayor distancia entre ellas a medida que incrementan los *epochs*, lo cual se confirma al permitir incrementar el hiperparámetro hasta donde el *early stop* detiene el entrenamiento. Se puede observar en la figura **7-17** que existe un error de generalización notable en el modelo, correspondiente a un modelo con sobre ajuste.

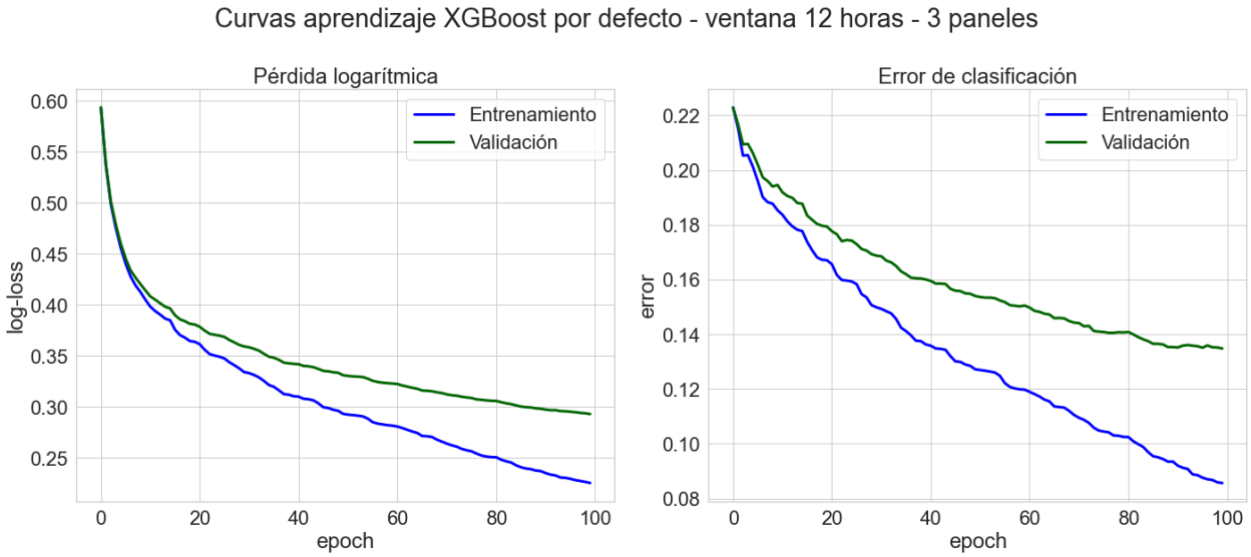


Figura 7-16: Curvas de aprendizaje modelo *XGBoost* ventana de 12 horas, 5 paneles, sin *early stopping*

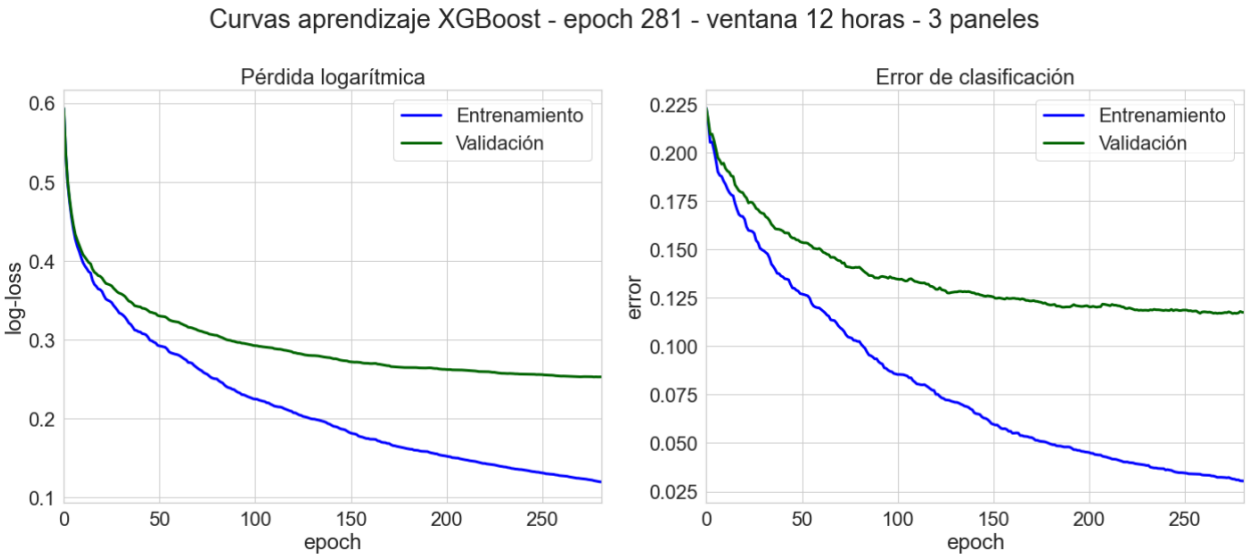


Figura 7-17: Curvas de aprendizaje modelo *XGBoost* ventana de 12 horas, 3 paneles, con *early stopping*

Con base en estos resultados se descartan los vecindarios de 3 paneles como modelos viables. Se seleccionan los modelos con base en vecindarios de 5 paneles como la opción más apropiada dentro de las combinaciones evaluadas en el diseño experimental. Respecto al ancho de ventana, considerando la similitud de los resultados para vecindarios de 5 paneles, se realiza un segundo tipo de validación para confirmar qué ancho de ventana permite obtener los mejores resultados.

7.5. Validación de modelos con datos de línea de tiempo completa

Para realizar una segunda prueba que permita evaluar la capacidad de detección de los modelos obtenidos en la sección 7.4 se selecciona el vecindario de termopares con mayor registro de fallos y se etiquetan manualmente todos los registros disponibles en la línea de tiempo. El vecindario se compone de 5 *Waffle Coolers B* entre las posiciones 40 y 44, como se muestra en la figura **7-18**.

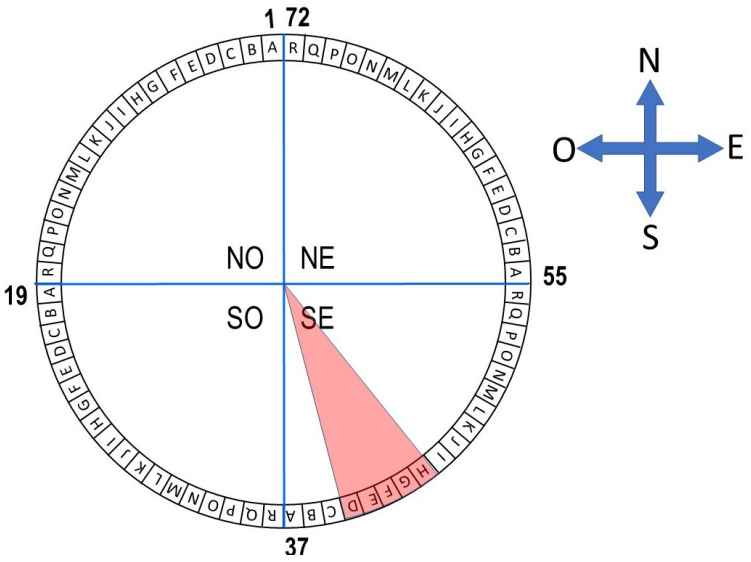


Figura 7-18: Vecindario de *Waffle Coolers* de validación

El *dataset* resultante tiene un total de 175,295 registros, de los cuales 50,724 contienen fallos (40.71 %). Este conjunto de datos no fue utilizado en el establecimiento de línea base debido a su elevado porcentaje de fallos que desbalancearía los datos para los *Plate Coolers*. Para efectos del presente capítulo se le denominará validación ciega a este conjunto de datos que no formó parte del proceso de entrenamiento y primera validación de los modelos.

7.5.1. Metodología

Se extraen los *labels* y *features* del *dataset* de validación ciega empleando ventanas de tiempo de los diferentes anchos mencionados en la figura **6-6**. Este nuevo conjunto de *features* es presentado a cada uno de los modelos previamente entrenados para que cada uno determine la existencia o inexistencia de fallos. Los resultados obtenidos de los modelos son contrastados con los *labels* existentes para establecer la matriz de consuón y métricas en cada caso.

Dado que los modelos se entrenaron para detectar un tipo específico de fallo y que utilizar la totalidad de la línea de tiempo abre la posibilidad a encontrar ventanas de tiempo ancladas o con outliers, la validación ciega se realiza siguiendo el diagrama de flujo mostrado en la figura **7-19**.

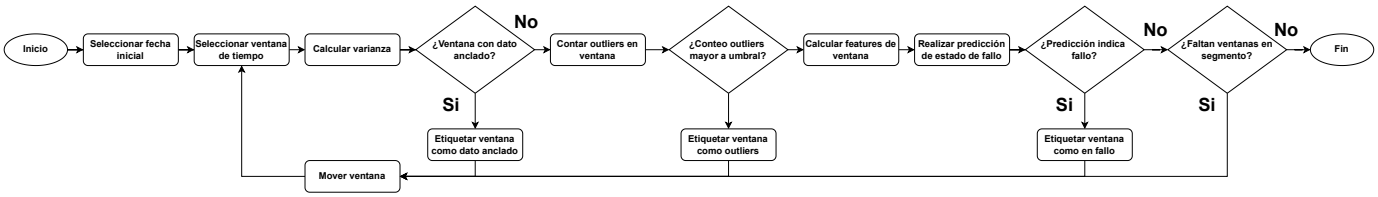


Figura 7-19: Diagrama de flujo de validación ciega

7.5.2. Resultados y discusión

Los resultados de cada modelo se presentan en la tabla **7-5**.

Tamaño de ventana horas	Sensibilidad	Especificidad	precisión
24	0.4980	0.7906	0.7059
12	0.5168	0.9243	0.8064
6	0.4794	0.9535	0.8163
3	0.5056	0.9016	0.7087
1	0.4627	0.9459	0.8061

Tabla 7-5: Resultados validación ciega

Se hace evidente un problema de sensibilidad de todos los modelos, es decir de su capacidad para identificar adecuadamente la existencia de un Verdadero positivo y poder diferenciarlo de un Falso Negativo. Al estudiar en detalle los registros que los modelos identificaron como **en fallo** se encontraron casos como el mostrado en la figura **7-20**.

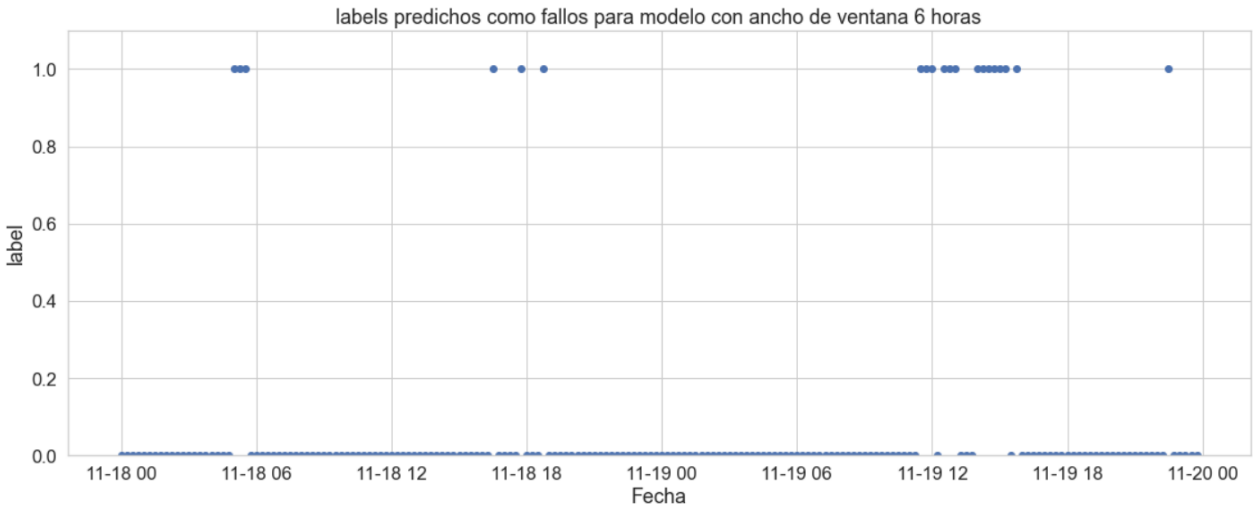


Figura 7-20: Extracto de *labels* etiquetados con modelo de ancho de ventana de 6 horas

Cada punto en la gráfica corresponde a un registro en el dataset de *labels* predichos por el modelo. Se observa la detección de fallos (*labels* con valor 1) de manera intermitente alrededor de las fechas como el marcador **11-18 18** o el marcador **11-19 12**. El comportamiento deseado es que los fallos detectados se presenten como una sucesión de *labels* con valor 1 a lo largo del intervalo en fallo y vuelvan a valor 0 cuando el fallo deje de afectar el vecindario de termopares.

Para reducir el efecto de estos valores intermitentes se propone filtrar el *dataset* de *labels* predichos empleando un contador *up-down* adaptado a partir del desarrollo de Ozdemir *et. al* [81].

7.6. Filtrado mediante contador up-down

El contador *up-down* es un algoritmo de umbralización diseñado para incrementar la probabilidad de detección a partir de la modificación de los tiempos de detección de una alarma a partir de un *trade-off* entre capacidad de detección y tasa de falsos positivos.

El contador *up-down* evalúa cada registro en una línea de tiempo e incrementa el valor de un contador interno si el valor leído se encuentra en los umbrales de alarma; el contador *up-down* enciende su señal de alarma si el contador interno alcanza su valor de umbral alto y mantendrá su señal de alarma encendida hasta que se lean suficientes registros fuera de los umbrales de alarma como para que el valor del contador interno caiga hasta el valor de umbral bajo. Los valores de umbrales y la forma como el contador interno crece y decae es determinado por quien implementa el contador *up-down*.

7.6.1. Metodología

Se diseña una versión reducida del contador *up-down* con valor de umbral alto variable, valor de umbral bajo de 0 y tanto incremento como detrimento lineal del contador interno, sumando o restando 1 unidad al contador con cada *label* en fallo o sin fallo, respectivamente. El funcionamiento del contador *up-down* reducido se ilustra en la figura **7-21**.

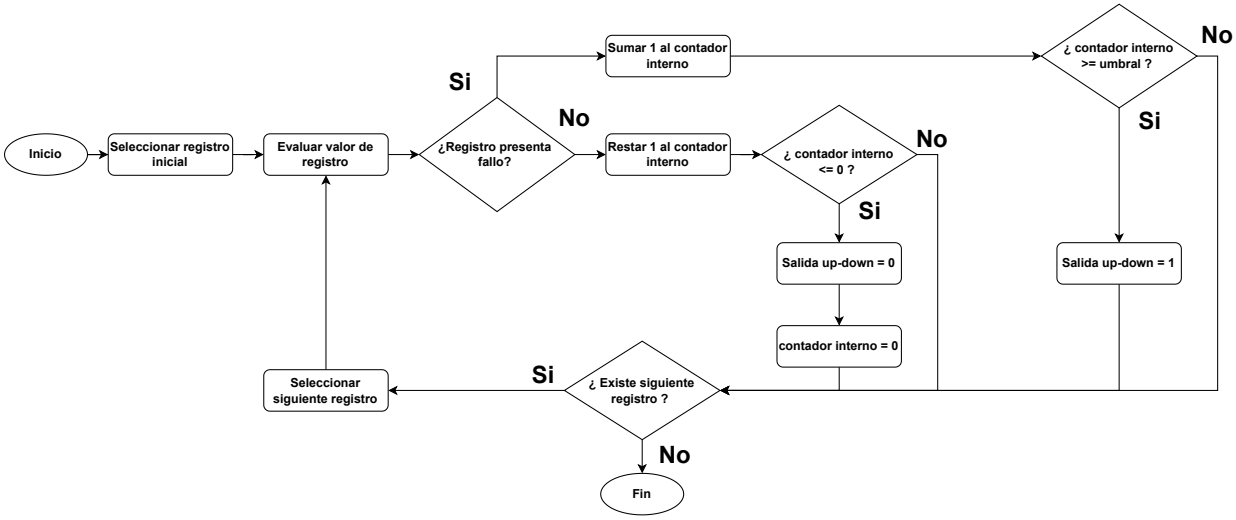


Figura 7-21: Diagrama de flujo de algoritmo de contador *up-down* reducido

Se realiza un barrido de parámetro del valor umbral alto del contador *up-down*, se filtran los diferentes conjuntos de *labels* predichos en la tabla **7-5** y se evalúa si existe algún cambio en el desempeño de la detección de fallos al agregar el contador *up-down* después de la predicción de los modelos *XGBoost*.

Para tener una segunda validación ciega, se etiquetó manualmente un vecindario de *Plate Coolers A* desde la posición angular 24 hasta la 32. Se generó adicionalmente un segundo tipo de dataset de features que incluye la pendiente de la ventana de datos, para evaluar la incidencia de este *feature* en los modelos de *Machine Learning*.

7.6.2. Resultados y discusión

Los resultados se sintetizan en las figuras **7-22** y **7-23**. Se hace la aclaración que un valor de filtro *up-down* de 1 equivale a no usar el filtro.

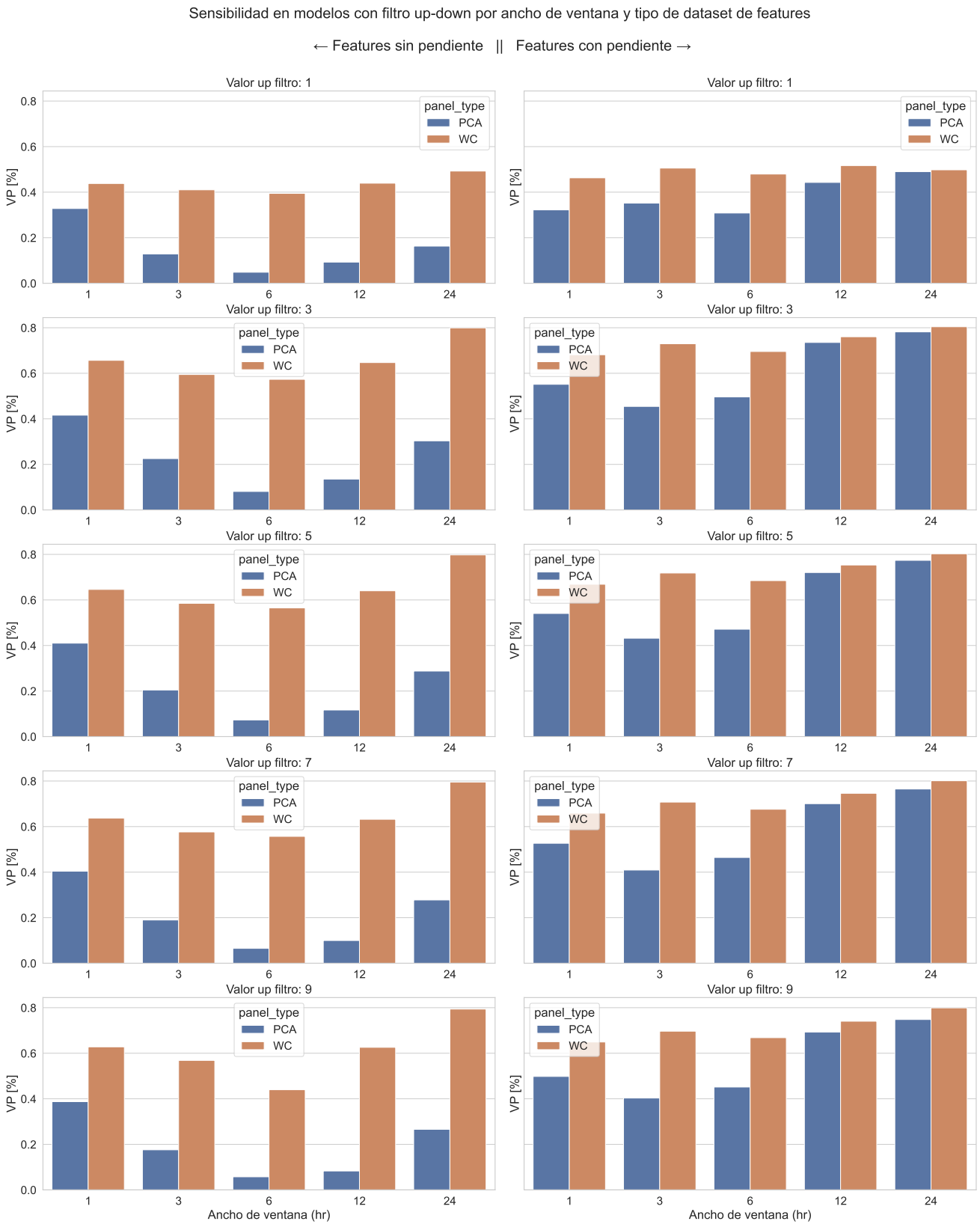


Figura 7-22: Cambio en sensibilidad para modelos de *Machine Learning* con filtros *up-down*

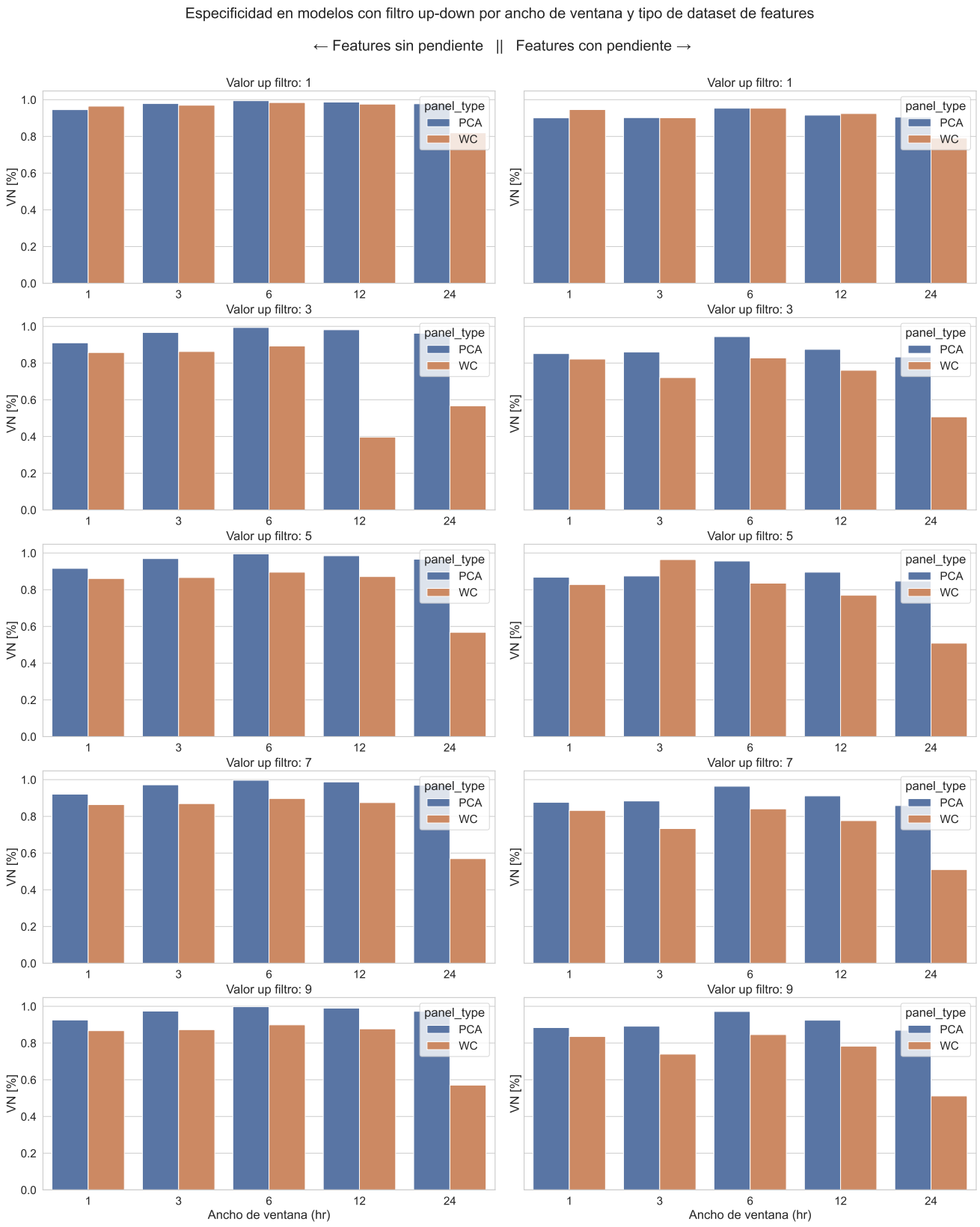


Figura 7-23: Cambio en especificidad para modelos de *Machine Learning* con filtros *up-down*

Se observa una mejoría notable en la sensibilidad, subiendo hasta valores alrededor de 80 % para los modelos con ancho de ventana de 24 horas. En cuanto a la especificidad se aprecia una ligera disminución, consecuente con el *trade-off* que proponer el contador *up-down*.

Adicionalmente se aprecia que la inclusión de la pendiente como *feature* permitió obtener valores más altos de sensibilidad, a diferencia de lo observado en la especificidad donde se observa una ligera reduccioón en esta métrica para los *features* con pendiente.

De las combinaciones observadas en las gráficas se identifica el modelo con ancho de ventana de 12 horas, *features* con pendiente y filtro *up-down* con umbral de 5 como el que presenta la combinación más alta de sensibilidad y especificidad, como se muestra en la tabla **7-6**.

umbral *up-down*	**Tipo panel**	**Sensibilidad**	**Especificidad**	**precisión**
3	**Plate Coolers A**	0.7351	0.8746	0.8738
	Waffle Coolers	0.7603	0.7605	0.7605
5	**Plate Coolers A**	0.7203	0.8953	0.8942
	Waffle Coolers	0.7533	0.7694	0.7650

Tabla 7-6: Comparación resultados de modelos con filtro *up-down*

Se observa que los resultados del modelo con valor umbral de 5 es ligeramente superior, por lo que se selecciona como el mejor candidato dentro del alcance de los experimentos realizados en el presente trabajo.

7.7. Aislamiento de termopar con fallo

Gracias a la arquitectura construida es posible detectar fallos en vecindarios de termopares con una tasa de sensibilidad por encima del 72 %, de acuerdo a las dos validaciones ciegas realizadas. Una vez detectado el vecindario en fallo, se puede emplear una métrica derivada de la varianza de cada panel para aislar la fuente del fallo.

Se toma como ejemplo dos de los puntos identificados como fallo en las pruebas de validación ciega. La primera ventana corresponde inicia en el marcador de tiempo **2014-11-16 10:00:00** para el vecindario de validación ciega conformado por *Waffle Coolers B*. La figura **7-24** muestra la línea de tiempo de la ventana detectada.

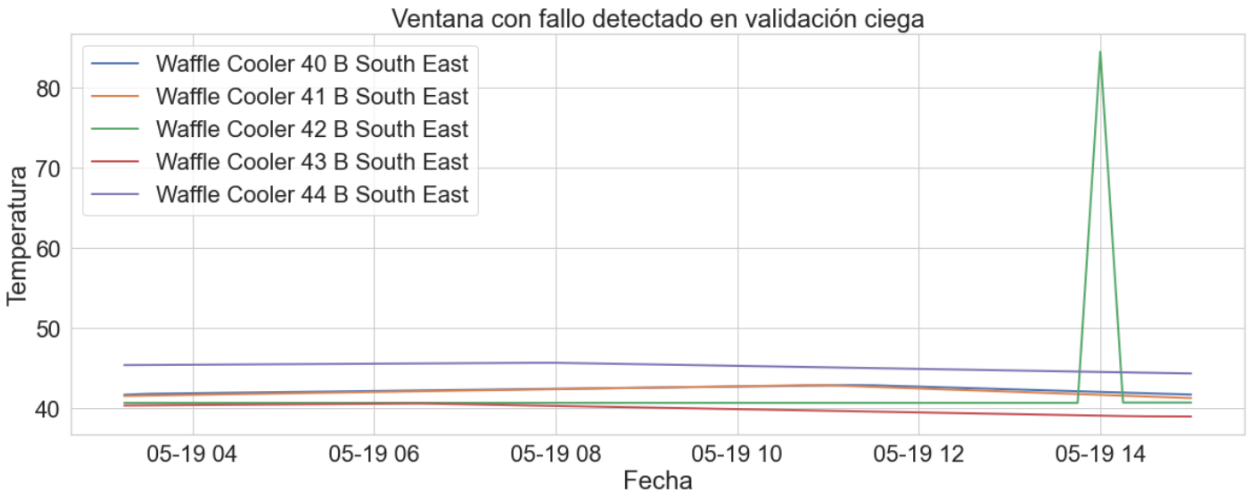


Figura 7-24: Ventana en fallo 1 detectada en validación ciega

Se observa que el fallo ocurre en el *Waffle Cooler 42 B*. Para aislar la fuente del fallo se calcula la varianza de cada señal, se calcula la mediana del conjunto de varianzas y posteriormente se calcula la distancia de cada varianza respecto a la mediana. El termopar cuya varianza muestre mayor distancia respecto a la mediana será el termopar con fallo, sin importar qué valor de varianza presenten los termopares libres de fallo. Se obtienen los resultados mostrados en la tabla **7-7** para el segmento de *Waffle Coolers*.

Termopar	Varianza	Distancia a la mediana
Waffle Cooler 40 B	0.1214	0.0737
Waffle Cooler 41 B	0.1951	0.0000
Waffle Cooler 42 B	40.000	39.804
Waffle Cooler 43 B	0.2972	0.1021
Waffle Cooler 44 B	0.1673	0.0278

Tabla 7-7: Varianza ventana en fallo 1 detectada en validación ciega

Se encuentra en este primer caso que el panel con fallo dentro del vecindario es el *Waffle Cooler 42 B*.

Un segundo caso se desarrolla a partir de una ventana detectada con fallo en el vecindario de *Plate Coolers A* de la validación ciega. Los valores de cada termopar en la ventana se muestran en la figura **7-25**.

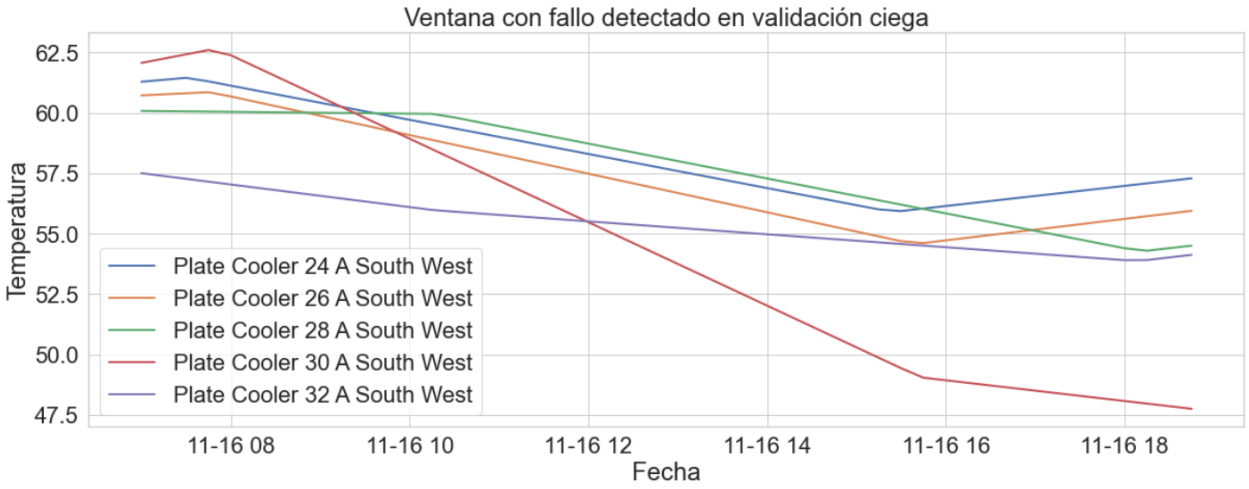


Figura 7-25: Ventana en fallo 2 detectada en validación ciega

En este caso se observa que el fallo ocurre en el *Plate Cooler 30 A*. Al calcular la varianza de cada termopar y la distancia respecto a la mediana del vecindario se obtienen los resultados mostrados en la tabla **7-8**.

Termopar	Varianza	Distancia a la mediana
Plate Cooler 24 A	3.2228	1.0188
Plate Cooler 26 A	4.4127	0.1711
Plate Cooler 28 A	4.2416	0.0000
Plate Cooler 30 A	26.694	22.4524
Plate Cooler 32 A	1.1215	3.1201

Tabla 7-8: Varianza ventana en fallo 2 detectada en validaci\'on ciega

Se identifica que el termopar con fallo es el *Plate Cooler 30 A* al mostrar la mayor distancia de su varianza respecto a la mediana del vecindario.

Conclusiones

- La combinación de criterios estadísticos y heurísticos permitió establecer un conjunto de datos enfocado en desarrollar un proceso de *FDI* para la red de termopares del horno.
- A partir de métricas como la matriz de correlación, varianza, normas hurísticas, análisis en ventanas de tiempo, observación y etiquetado manual fue posible establecer una línea base tanto para el entrenamiento de modelos de *Machine Learning* dentro del alcance del presente documento, como para el entrenamiento y validación de modelos que puedan desarrollarse en trabajos futuros sobre este mismo conjunto de datos.
- El desarrollo del modelo a partir de diseños experimentales combinados permitió establecer una metodología para la efectiva detección y aislamiento de fallos en la red de termopares usando técnicas basadas en datos.
- De las combinaciones evaluadas en el diseño experimental, se obtiene el modelo con el mayor desempeño en cuanto a sensibilidad y especificidad a partir de los siguientes parámetros:
  - Modelo *XGBoost* entrenado con 499 *epochs*
  - Arquitectura de vecindarios **modelo único**
  - Ancho de ventana de 12 horas
  - Vecindarios de 5 paneles
  - Contador *up-down* con umbral de 5
  - Dataset de features por ventana compuesto por:
    1. Correlación entre paneles
    2. Varianza de cada panel
    3. Mediana de cada panel
    4. Pendiente de cada panel
- La realización de pruebas de validación ciegas permitió identificar falencias en los modelos de *Machine Learning* que no se evidenciaron durante el proceso de validación inicial debido a que los datos de entrenamiento y validación son cuidadosamente seleccionados para que reflejen claramente los comportamientos que se quieren clasificar.

- Mediante la combinación del modelo de *Machine Learning* de detección y la comparación de la distancia de las varianzas respecto a la mediana de varianzas, es posible detectar y aislar efectivamente el termopar en fallo del vecindario etiquetado por el modelo como con presencia de fallo.

Financiación y privacidad de la información:

El presente trabajo fue realizado dentro del marco de la colaboración entre la Universidad Nacional de Colombia y Cerro Matoso S.A, financiada por el Ministerio Colombiano de Ciencia mediante la convocatoria 786: *"Convocatoria para el registro de proyectos que aspiran a obtener beneficios tributarios por inversión en CTel"*.

La totalidad de los registros empleados en el presente proyecto son de carácter privado y pertenecen a Cerro Matoso S.A. Dichos registros no pueden ser publicados, compartidos o reproducidos total o parcialmente sin el conocimiento y expresa autorización de Cerro Matoso S.A.

Bibliografía

[1] CMSA, *PR032018OP - Manual del Sistema de Control Estructural del Horno Eléctrico 412-FC-01*, 02 ed., 2017.

[2] L. Shen, Z. Jiang, W. Gui, C. Yang, Y. Wang, and B. Sun, "Modelling of inner surface temperature field of blast furnace wall based on inverse heat conduction problems," *IFAC-PapersOnLine*, vol. 52, pp. 78–83, 2019.

[3] D. A. Tibaduiza, J. León, L. Bonilla, B. Rueda, O. Zurita, J. C. Forero, J. Vitola, D. Segura, E. Forero, and M. Anaya, "GAP MONITORING IN REFURBISHMENT TASKS IN a FERRONICKEL FURNACE AT CERRO MATOSO SA," in *XI International Conference on Structural Dynamics*, EASD, 2020.

[4] D. A. Tibaduiza, J. X. Leon-Medina, R. Gomez, J. Ricardo, B. Rueda, O. Zurita, and J. C. Forero, "Structural health monitoring system for furnace refractory wall thickness measurements at cerro matoso sa," in *European Workshop on Structural Health Monitoring* (P. Rizzo and A. Milazzo, eds.), (Cham), pp. 414–423, Springer International Publishing, 2021.

[5] B. Zhou, H. Ye, H. Zhang, and M. Li, "Process monitoring of iron-making process in a blast furnace with PCA-based methods," *Control Engineering Practice*, vol. 47, no. February 2016, pp. 1–14, 2016.

[6] A. Van Herwaarden and P. Sarro, "Thermal sensors based on the seebeck effect," in *Sensors and Actuators*, vol. 10, pp. 321–346, 1986.

[7] I. E. Comission, "IEC 60584-1:2013 - Thermocouples - Part 1: EMF specifications and tolerances," International Standard, International Electrotechnical Commission, 08 2013.

[8] R. X. Guo, K. Guo, and J. K. Dong, "Fault diagnosis for sensors in a class of nonlinear systems," *IMA Journal of Mathematical Control and Information*, vol. 35, no. 2, pp. 375–391, 2018.

[9] T. Zhang, H. Ye, W. Wang, and H. Zhang, "Fault diagnosis for blast furnace ironmaking process based on two-stage principal component analysis," *ISIJ International*, vol. 54, no. 10, pp. 2334–2341, 2014.

[10] P. A. Samara, G. N. Fouskitakis, J. S. Sakallariou, and S. D. Fassois, "A statistical method for the detection of sensor abrupt faults in aircraft control systems," *IEEE Transactions on Control Systems Technology*, vol. 16, no. 4, pp. 789–798, 2008.

[11] Y. Wang, Y. Yan, and Q. Wang, "Wavelet-Based Feature Extraction in Fault Diagnosis for Biquad High-Pass Filter Circuit," *Mathematical Problems in Engineering*, vol. 2016, 2016.

[12] X. Dai, F. Qin, Z. Gao, K. Pan, and K. Busawon, "Model-based on-line sensor fault detection in Wireless Sensor Actuator Networks," *Proceeding - 2015 IEEE International Conference on Industrial Informatics, INDIN 2015*, no. 1, pp. 556–561, 2015.

[13] P. R. J, "Using Analytical Redundancy," *Computing & Control Engineering*, vol. 8, no. May, pp. 127–136, 1991.

[14] R. Shahnazi and Q. Zhao, "Adaptive Fuzzy Descriptor Sliding Mode Observer-based Sensor Fault Estimation for Uncertain Nonlinear Systems," *Asian Journal of Control*, vol. 18, no. 4, pp. 1478–1488, 2016.

[15] A. S. Willsky and H. L. Jones, "A Generalized Likelihood Ratio Approach to the Detection and Estimation of Jumps in Linear Systems," *IEEE Transactions on Automatic Control*, vol. 21, no. 1, pp. 108–112, 1976.

[16] M. A. Djeziri, A. Aitouche, and B. O. Bouamama, "Sensor fault detection of energetic system using modified parity space approach," *Proceedings of the IEEE Conference on Decision and Control*, pp. 2578–2583, 2007.

[17] H. Tolouei and M. A. Shoorehdeli, "Nonlinear parity approach to sensor fault detection in pH neutralization system," *2017 25th Iranian Conference on Electrical Engineering, ICEE 2017*, no. Icee, pp. 889–894, 2017.

[18] Y. Cheng, M. D'Arpino, and G. Rizzoni, "Fault diagnosis in lithium-ion battery of hybrid electric aircraft based on structural analysis," in *2022 IEEE Transportation Electrification Conference & Expo (ITEC)*, pp. 997–1004, 2022.

[19] H. Shahnazari, P. Mhaskar, J. House, and T. Salsbury, "Fault diagnosis design for heating, ventilation and air conditioning systems," in *2018 Annual American Control Conference (ACC)*, pp. 5787–5792, 2018.

[20] D. Chakraborty and H. Elzarka, "Early detection of faults in HVAC systems using an XGBoost model with a dynamic threshold," *Energy and Buildings*, vol. 185, pp. 326–344, 2019.

[21] K. Pal and J. A. Campbell, "An Application of Rule-Based and Case-Based Reasoning within a Single Legal Knowledge- Based System," *Data Base for Advances in Information Systems*, vol. 28, no. 4, pp. 48–63, 1997.

[22] Z. Dong, J. Zhao, J. Duan, M. Wang, and H. Wang, "Research on Agricultural Machinery Fault Diagnosis System Based on Expert System," *Proceedings of 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference, IMCEC 2018*, no. Imcec, pp. 2057–2060, 2018.

[23] L. A. Zadeh, "Fuzzy Probabilisties and Their Role in Decision Analysis.," *IFAC Proceedings Volumes*, vol. 15, no. 1, pp. 15–21, 1982.

[24] K. Raja and R. Leichombam, "Distributed Fault Node Detection and Classification using Fuzzy Logic and Management Scheme for Wireless Sensor Networks," *2018 10th International Conference on Advanced Computing, ICoAC 2018*, pp. 292–298, 2018.

[25] W. He, P. L. Qiao, Z. J. Zhou, G. Y. Hu, Z. C. Feng, and H. Wei, "A New Belief-Rule-Based Method for Fault Diagnosis of Wireless Sensor Network," *IEEE Access*, vol. 6, pp. 9404–9419, 2018.

[26] Jian-Bo Yang, Jun Liu, Jin Wang, How-Sing Sii, and Hong-Wei Wang, "Belief rule-base inference methodology using the evidential reasoning approach-rimer," *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, vol. 36, no. 2, pp. 266–285, 2006.

[27] X. Yuan, M. J. Liebelt, P. Shi, and B. J. Phillips, "Development of Rule-Based Agents for Autonomous Parking Systems by Association Rules Mining," *Proceedings - International Conference on Machine Learning and Cybernetics*, vol. 2019-July, pp. 1–6, 2019.

[28] Maciol, Andrzej, Jedrusik, Stanislaw, and Palinski, Andrzej, "How to increase effectiveness of inference in rule-based systems," *Proceedings - 2019 2nd International Conference on Artificial Intelligence for Industries, AI4I 2019*, pp. 107–110, 2019.

[29] Z. Wang, Y. Zhou, and G. Li, "Anomaly detection by using streaming k-means and batch k-means," in *2020 5th IEEE International Conference on Big Data Analytics (ICBDA)*, pp. 11–17, 2020.

[30] K. Tutuncu, I. Cinar, R. Kursun, and M. Koklu, "Edible and poisonous mushrooms classification by machine learning algorithms," in *2022 11th Mediterranean Conference on Embedded Computing (MECO)*, pp. 1–4, 2022.

[31] T. Badriyah, N. Sakinah, I. Syarif, and D. R. Syarif, "Machine learning algorithm for stroke disease classification," in *2020 International Conference on Electrical, Communication, and Computer Engineering (ICECCE)*, pp. 1–5, 2020.

[32] Y. Sönmez, T. Tuncer, H. Gökal, and E. Avci, "Phishing web sites features classification based on extreme learning machine," in *2018 6th International Symposium on Digital Forensic and Security (ISDFS)*, pp. 1–5, 2018.

[33] K. Pal and B. V. Patel, "Automatic multiclass document classification of hindi poems using machine learning techniques," in *2020 International Conference for Emerging Technology (INCET)*, pp. 1–5, 2020.

[34] M. N. Hasan and I. Koo, "Machine learning-based sensor drift fault classification using discrete cosine transform," in *2021 International Conference on Electronics, Communications and Information Technology (ICECIT)*, pp. 1–4, 2021.

[35] M. Kubat, *An introduction to machine learning.* Springer International Publishing, 2017.

[36] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, "A survey on distributed machine learning," *ACM Computing Surveys*, vol. 53, 6 2020.

[37] G. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM SIGKDD Explorations Newsletter*, pp. 20–29, 6 2004.

[38] N. Japkowicz and S. Stephen, "The class imbalance problem: A systematic study," *Intelligent data analysis*, vol. 6, no. 5, pp. 429–449, 2002.

[39] K. Pal and B. V. Patel, "Data classification with k-fold cross validation and holdout accuracy estimation methods with 5 different machine learning techniques," in *2020 Fourth International Conference on Computing Methodologies and Communication (ICCMC)*, pp. 83–87, 2020.

[40] F. K. Alarfaj, I. Malik, H. U. Khan, N. Almusallam, M. Ramzan, and M. Ahmed, "Credit card fraud detection using state-of-the-art machine learning and deep learning algorithms," *IEEE Access*, vol. 10, pp. 39700–39715, 2022.

[41] G. M. Suhas Jain, N. Rakesh, K. Pranavi, and L. Bale, "A novel approach in credit card fraud detection system using machine learning techniques," in *2021 International Conference on Forensics, Analytics, Big Data, Security (FABS)*, vol. 1, pp. 1–5, 2021.

[42] C. Srinilta and S. Kanharattanachai, "Application of natural neighbor-based algorithm on oversampling smote algorithms," in *2021 7th International Conference on Engineering, Applied Sciences and Technology (ICEAST)*, pp. 217–220, 2021.

[43] S. Sehra, D. Flores, and G. D. Montañez, "Undecidability of underfitting in learning algorithms," in *2021 2nd International Conference on Computing and Data Science (CDS)*, pp. 591–594, 2021.

[44] J. Camacho-Olarte, J. E. S. Torres, D. A. G. Jimenez, J. X. L. Medina, R. C. G. Vargas, D. A. V. Cardenas, C. Gutierrez-Osorio, B. Rueda, W. Vargas, D. A. T. Burgos, C. A. P. Bonilla, J. S. Esmeral, and F. Restrepo-Calle, "A data cleaning approach for a structural health monitoring system in a 75 mw electric arc ferronickel furnace," *Engineering Proceedings*, vol. 2, no. 1, 2020.

[45] U. N. de Colombia, "Producto 2.6 informe técnico del análisis de correlación y redundancia física de los waffle coolers y plate coolers del horno línea 1 fc01," Informe Técnico 2.6, Universidad Nacional de Colombia, 2020.

[46] R. Cao, Y. Chen, M. Shen, J. Chen, J. Zhou, C. Wang, and W. Yang, "A simple method to improve the quality of ndvi time-series data by integrating spatiotemporal information with the savitzky-golay filter," *Remote Sensing of Environment*, vol. 217, pp. 244–257, 2018.

[47] J. Chen, P. Jönsson, M. Tamura, Z. Gu, B. Matsushita, and L. Eklundh, "A simple method for reconstructing a high-quality ndvi time-series data set based on the savitzky-golay filter," *Remote Sensing of Environment*, vol. 91, pp. 332–344, 6 2004.

[48] S. Mandal, B. Santhi, S. Sridhar, K. Vinolia, and P. Swaminathan, "Minor fault detection of thermocouple sensor in nuclear power plants using time series analysis," *Annals of Nuclear Energy*, vol. 134, pp. 383–389, 2019.

[49] V. N. Vapnik, *Statistical learning theory.* Wiley, 1998.

[50] L. Breiman, "Bagging predictors," in *Machine Learning*, vol. 24, pp. 123–140, Springer Nature, 1996.

[51] L. Breiman, "Random forests," in *Machine Learning*, vol. 45, pp. 5–32, Springer Nature, 2001.

[52] P. Mekha and N. Teeyasuksaet, "Image classification of rice leaf diseases using random forest algorithm," in *2021 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering*, pp. 165–169, 2021.

[53] S. Li and Y. Yan, "Fault arc detection based on time and frequency domain analysis and radom forest," in *2021 International Conference on Computer Network, Electronic and Automation (ICCNEA)*, pp. 248–252, 2021.

[54] Q. Yao, J. Wang, L. Yang, H. Su, and G. Zhang, "A fault diagnosis method of engine rotor based on random forests," in *2016 IEEE International Conference on Prognostics and Health Management (ICPHM)*, pp. 1–4, 2016.

[55] R. Fezai, K. Dhibi, M. Mansouri, M. Trabelsi, M. Hajji, K. Bouzrara, H. Nounou, and M. Nounou, "Effective random forest-based fault detection and diagnosis for wind energy conversion systems," *IEEE Sensors Journal*, vol. 21, no. 5, pp. 6914–6921, 2021.

[56] J. X. Leon-Medina, J. Camacho-Olarte, B. Rueda, W. Vargas, L. Bonilla, J. Ruiz, J. Sofrony, J. A. Guerra-Gomez, F. Restrepo-Calle, and D. A. Tibaduiza, "Monitoring of the refractory lining in a shielded electric arc furnace: An online multitarget regression trees approach," *Structural Control and Health Monitoring*, vol. 29, Nov. 2021.

[57] A. N. Warrier and S. Amuru, "How to choose a neural network architecture? - a modulation classification example," in *2020 IEEE 3rd 5G World Forum (5GWF)*, pp. 413–417, 2020.

[58] R. Kanagaraj, E. Elakiya, N. Rajkumar, K. Srinivasan, and S. Sriram, "Fuzzy neural network classification model for multi labeled electricity consumption data set," in *2022 4th International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pp. 1037–1041, 2022.

[59] D. Prabhu, M. Abedin, W. Winfree, and E. Madaras, "Disbond detection through ultrasonic signal classification using an artificial neural network," in *IJCNN-91-Seattle International Joint Conference on Neural Networks*, vol. ii, pp. 906 vol.2–, 1991.

[60] F. A. Uçkun, H. Özer, E. Nurbas, and E. Onat, "Direction finding using convolutional neural networks and convolutional recurrent neural networks," in *2020 28th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, 2020.

[61] R. Xin, J. Zhang, and Y. Shao, "Complex network classification with convolutional neural network," *Tsinghua Science and Technology*, vol. 25, no. 4, pp. 447–457, 2020.

[62] J. X. León-Medina, J. Camacho, C. Gutierrez-Osorio, J. E. Salomón, B. Rueda, W. Vargas, J. Sofrony, F. Restrepo-Calle, C. Pedraza, and D. Tibaduiza, "Temperature prediction using multivariate time series deep learning in the lining of an electric arc furnace for ferronickel production," *Sensors*, vol. 21, no. 20, 2021.

[63] D. F. Godoy-Rojas, J. X. Leon-Medina, B. Rueda, W. Vargas, J. Romero, C. Pedraza, F. Pozo, and D. A. Tibaduiza, "Attention-based deep recurrent neural network to forecast the temperature behavior of an electric arc furnace side-wall," *Sensors*, vol. 22, no. 4, 2022.

[64] C. Cortes, V. Vapnik, and L. Saitta, "Support-vector networks," *Machine Leaming*, vol. 20, pp. 273–297, 1995.

[65] L. Devroye, L. Györfi, and G. Lugosi, *Vapnik-Chervonenkis Theory*, pp. 187–213. New York, NY: Springer New York, 1996.

[66] G. Xu, Z. Cao, B.-G. Hu, and J. C. Principe, "Robust support vector machines based on the rescaled hinge loss function," *Pattern Recognition*, vol. 63, pp. 139–148, 2017.

[67] Z. Chen, L. Hong, Y. Gu, M. Wu, and Z. Yan, "A novel support vector machine multi-classification strategy for power transformer fault diagnosis," in *2022 IEEE 5th International Electrical and Energy Conference (CIEEC)*, pp. 3368–3373, 2022.

[68] X. Zhang, S. Wang, and Y. Zhao, "Application of support vector machine and least squares vector machine to freight volume forecast," in *2011 International Conference on Remote Sensing, Environment and Transportation Engineering*, pp. 104–107, 2011.

[69] J. Wang, L. Yu, H. Liu, and Y. Pang, "Analysis of power quality disturbance based on improved hht and bpso-svm," in *2022 IEEE 5th International Electrical and Energy Conference (CIEEC)*, pp. 748–753, 2022.

[70] R. Kumar, C. M. Sharma, V. M. Chariar, S. Hooda, and R. Beri, "Emotion analysis of news and social media text for stock price prediction using svm-lstm-gru composite model," in *2022 International Conference on Computational Intelligence and Sustainable Engineering Solutions (CISES)*, pp. 329–333, 2022.

[71] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, (New York, NY, USA), pp. 785–794, Association for Computing Machinery, 2016.

[72] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *The Annals of Statistics*, vol. 29, no. 5, pp. 1189–1232, 2001.

[73] L. Jidong and Z. Ran, "Dynamic weighting multi factor stock selection strategy based on xgboost machine learning algorithm," in *2018 IEEE International Conference of Safety Produce Informatization (IICSPI)*, pp. 868–872, 2018.

[74] J. Bao, "Multi-features based arrhythmia diagnosis algorithm using xgboost," in *2020 International Conference on Computing and Data Science (CDS)*, pp. 454–457, 2020.

[75] Z. Zheng, S. Pan, H. Luo, and Z. Guo, "Porosity prediction based on gs+ga-xgboost," in *2020 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)*, pp. 1014–1020, 2020.

[76] D. Zhang, L. Qian, B. Mao, C. Huang, B. Huang, and Y. Si, "A data-driven design for fault detection of wind turbines using random forests and xgboost," *IEEE Access*, vol. 6, pp. 21020–21031, 2018.

[77] B. Liu, X. Wang, K. Sun, J. Zhao, and L. Li, "Fault diagnosis of photovoltaic array based on xgboost method," in *2021 IEEE Sustainable Power and Energy Conference (iSPEC)*, pp. 3733–3738, 2021.

[78] C. Martínez Bencardino, *Estadística y muestreo*, p. 308. Colombia: ECOE Ediciones Ltda, 13 ed., 2012.

[79] M. Kubat, *Performance Evaluation*, pp. 211–228. Springer International Publishing, 2 ed., 2017.

[80] L. Prechelt, *Early Stopping — But When?*, pp. 53–67. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

[81] A. A. Ozdemir, P. Seiler, and G. J. Balas, "Wind turbine fault detection using counter-based residual thresholding," *IFAC Proceedings Volumes (IFAC-PapersOnline)*, vol. 44, pp. 8289–8294, 2011.