



Machine Learning – Chancador Primario

Desarrollo de **solución** que permita **alertar probabilidad** de **fallas** en **Chancador Primario** de Cerro Negro Norte en **tiempo real**, automatizar la **recolección de datos** operacionales y facilitar la visualización.



Agenda

- ▶ **Componentes de la solución.**
- ▶ **Etapas iniciales – Solución Machine Learning.**
- ▶ **Búsqueda de variables claves.**
- ▶ **Modelos Machine Learning (Entrenamiento, evaluación y solución).**
- ▶ **Demo WebApp.**
- ▶ **Próximos pasos.**



Entrenamiento de modelos de predicción. Serialización de modelos con formato .pkl



Extracción de datos.
Procesamiento y escalado de variables.
Predicción continua.



Implementación modelos predictivos (Potencia real, temperatura ...)

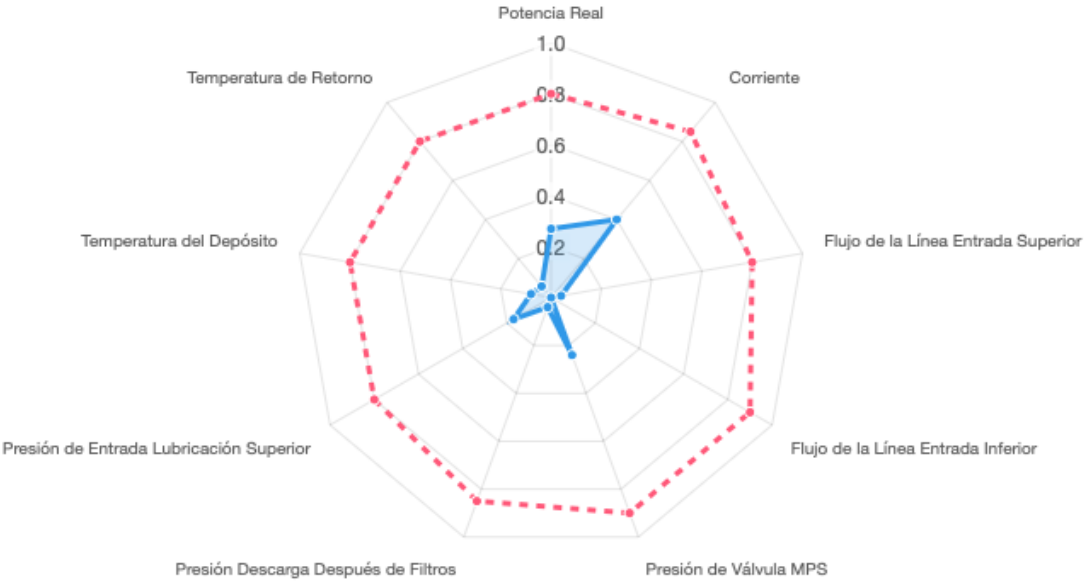


Ejecución de alertas cada 2 minutos



Diagrama de arquitectura de un sistema de procesamiento de datos en la nube de AWS. El sistema comienza con una API de PWEB que llama a un script 'chancado.py' y a un módulo 'c2 module' (ambos en un contenedor de Docker). El script 'chancado.py' envía datos a 'cargaipm10' (API de AWS IoT) y a 'chancadoCrm' (API de AWS IoT). 'cargaipm10' envía datos a 'et_ipm10PGST' (API de AWS IoT). 'chancadoCrm' envía datos a 'Salida/chancado.csv' (API de AWS IoT). Los datos se almacenan en S3 en buckets como 'modelo/chancado/' y 'salida/chancado/'. El sistema también incluye un 'Event 01:00' y un 'Repositorio'.

WebApp



Etapas iniciales



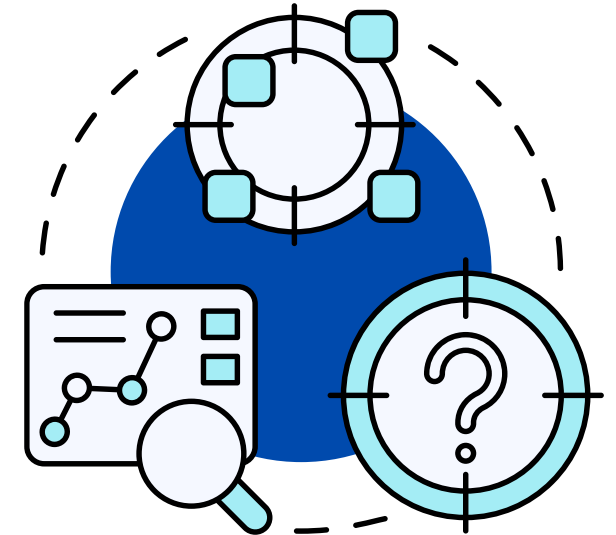
Recopilación de datos:

- Documentos técnicos
- Plano del chancador
- Valores límite de algunos sensores
- Listado de eventos del chancador
- Registros históricos de valores de distintos sensores



Procesamiento y limpieza de datos

- Unificación de datos históricos y estandarización de formatos.
- Filtro de datos operacionales.
- Normalización de datos para disminuir ruido en los datos.
- Cálculo de variables estadísticas.



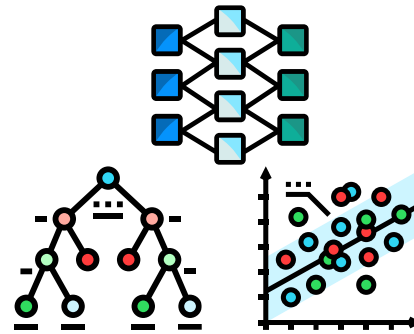
Análisis exploratorio

- Análisis de tendencias por sensor, con y sin filtro operacional.
- Evaluación de distribución de valores y desviaciones estándar.
- Detección visual de comportamientos anómalos previos a fallas.

Solución de Machine Learning



Entrada de nuevos registros



Procesamiento Machine Learning



Salida de probabilidades de falla

Modelos de Machine Learning a desarrollar:

- Redes neuronales (LSTM, CNN)
- Random Forest Classifier
- LSTM + Random Forest (Modelo Combinado)
- Autoencoders para detección de anomalías
- XGBoost
- Modelos supervisados con fallas sintéticas
- Detección de anomalías con variables combinadas

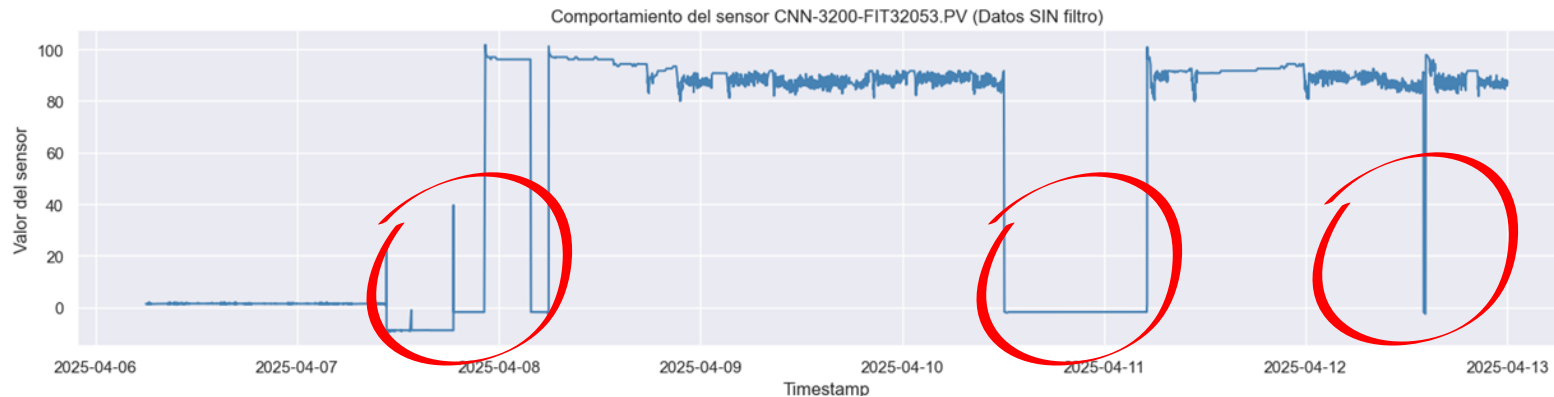
Problemas y desafíos a trabajar para aplicación de la solución:

- Sensores con alta variabilidad en su comportamiento
- Ausencia de sensores nuevos en datos históricos
- Comportamiento anómalo en sensores sin registro de falla en archivo de eventos
- Cambios en el comportamiento de los valores normales de algunos sensores

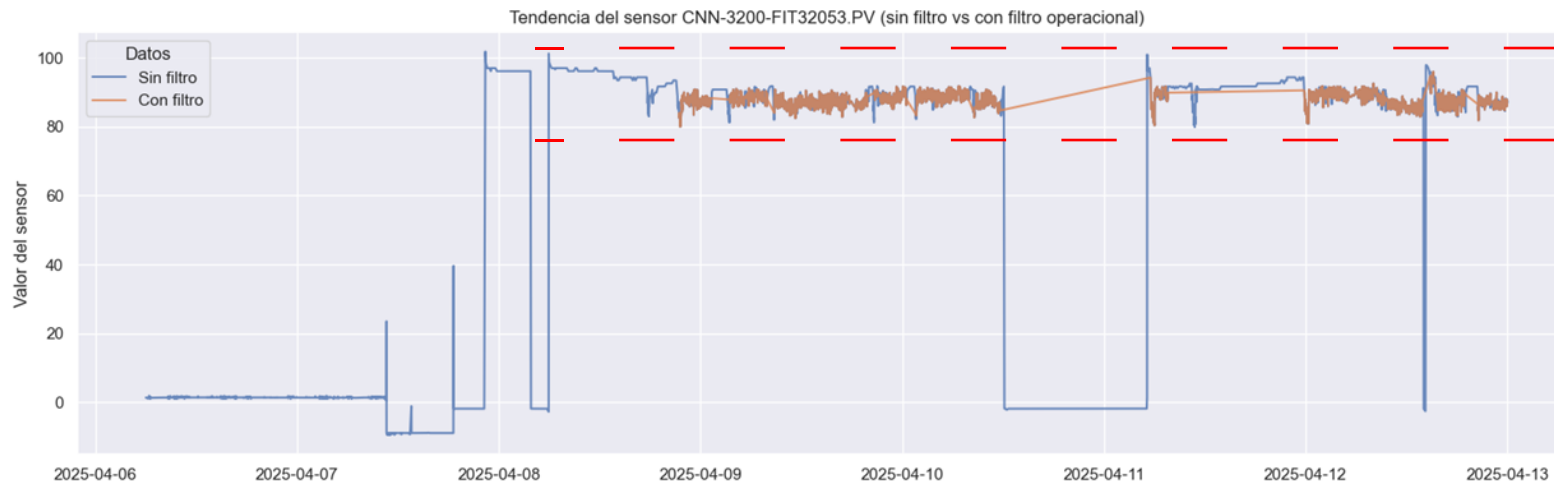
Selección de variables claves

Filtro de datos en estado operativo del chancador

En el análisis exploratorio se observan muchos casos de datos anómalos en el comportamiento de los sensores donde se acusa estados de mantenimiento o stand by del chancador. Para esto se considera excluir datos que tengan indicador de funcionamiento del motor principal en 0 y que tengan poco tonelaje de material de chancado.



Comportamiento anómalo de los datos



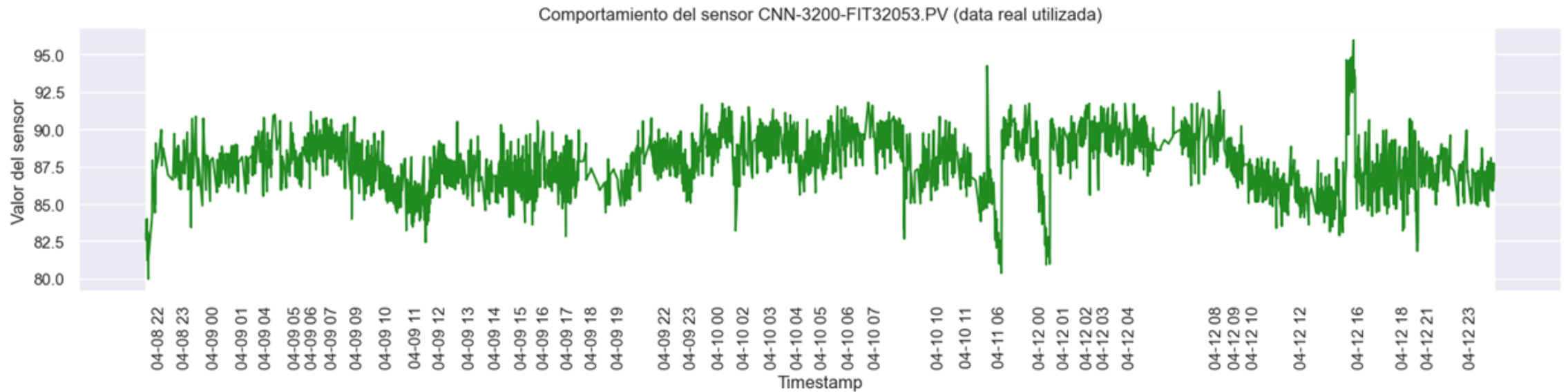
Mayor estabilidad en los datos

**Comparación de datos originales vs
datos en estado "operativo"**

Selección de variables claves

Resultado de filtro operacional:

Con este filtro se obtienen datos con mayor consistencia para el entrenamiento de los modelos de machine learning, permitiendo detectar con mayor precisión anomalías reales y no caer en exceso de falsos casos de falla.



Selección de variables claves

¿Qué criterios se utilizaron?

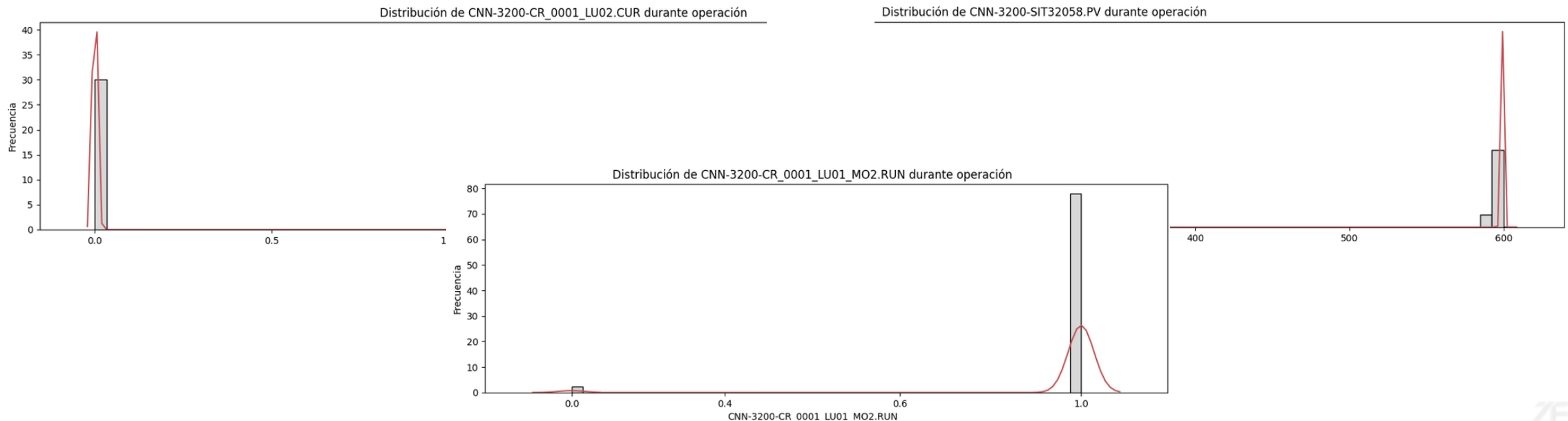
Se priorizaron sensores con comportamiento más estable, continuo y representativo de condiciones.

La selección se basó en:

- Análisis de correlación con eventos de falla.
- Evaluación de ruido y variabilidad de los datos

Exclusión de sensores o indicadores

Se excluyeron sensores o indicadores con distribución de valores de escaso aporte para un algoritmo de machine learning o con valores binarios como el caso de los indicadores de funcionamiento (".RUN")



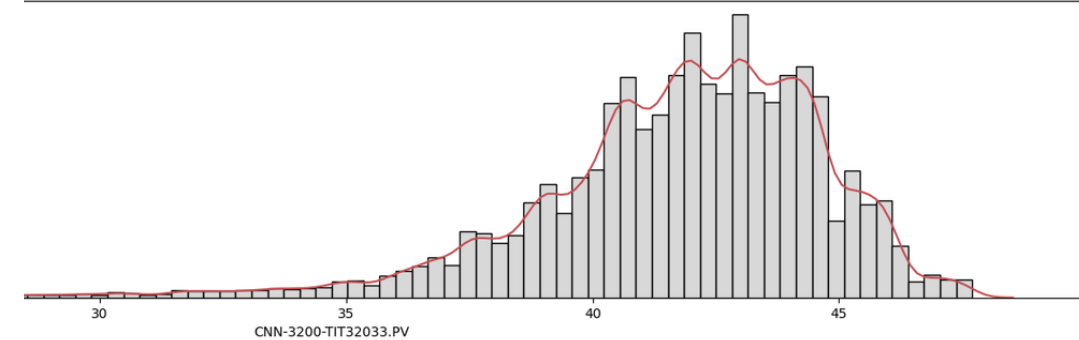
Selección de variables claves

Sensores e indicadores seleccionados:

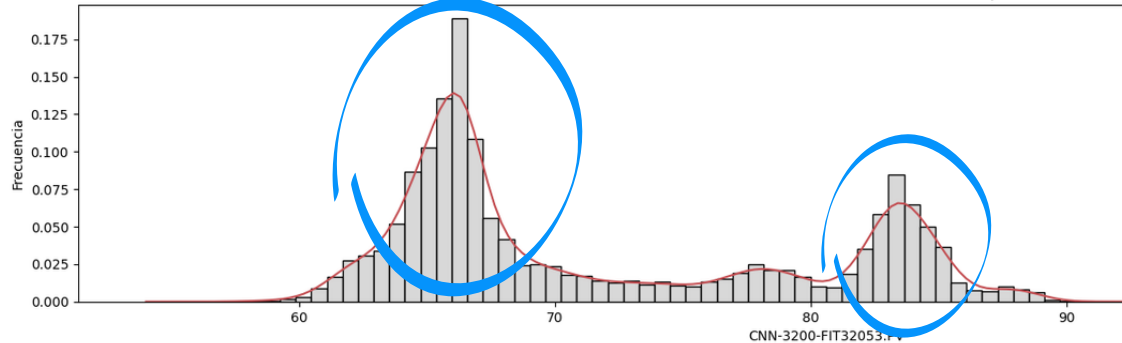
- Indicador de operación del motor principal (CNN-3200-CR_0001_MO.RUN)
- Tonelaje de material chancado (CNN-3200-WIC32149.PV)
- Potencia del motor principal (CNN-3200-CR_0001_MO.PWR)
- Corriente del motor principal (CNN-3200-CR_0001_MO.CUR)
- Flujo de entrada de lubricación superior (CNN-3200-FIT32053.PV)
- Flujo de entrada de lubricación inferior (CNN-3200-FIT32054.PV)
- Presión del cilindro de balance (CNN-3200-PIT32031.PV)
- Presión de descarga después de filtros (CNN-3200-PIT32043.PV)
- Presión de Línea de Lubricación Superior (CNN-3200-PIT32056.PV)
- Temperatura de depósito de aceite (CNN-3200-TIT32045.PV)
- Temperatura de aceite de retorno (CNN-3200-TIT32046.PV)

Utilizados para filtrar datos en operación

Distribución de CNN-3200-TIT32033.PV durante operación

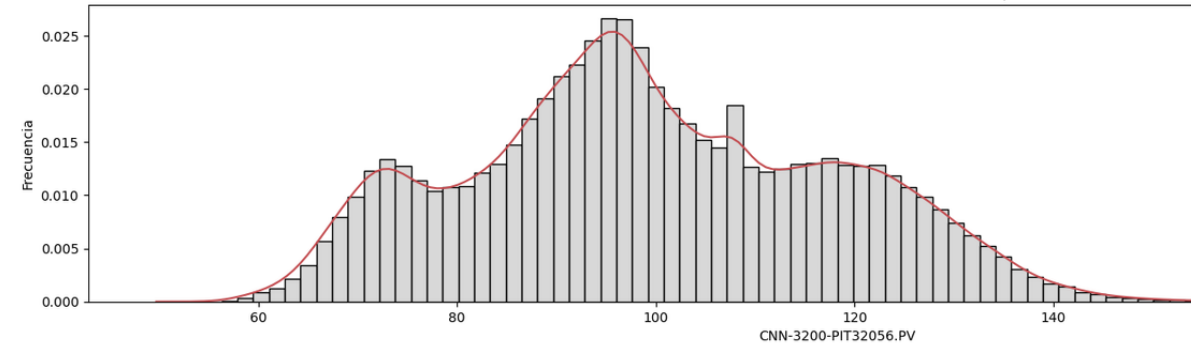


Distribución de CNN-3200-FIT32053.PV durante operación

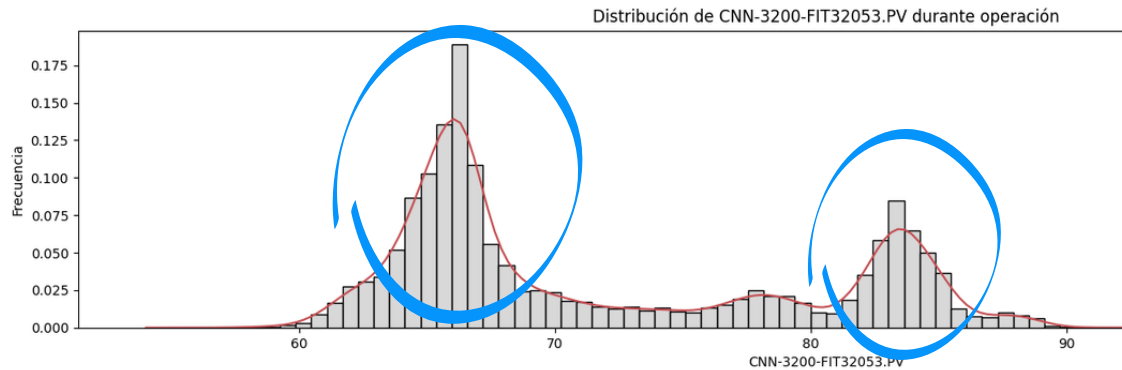


Comportamiento bimodal

Distribución de CNN-3200-PIT32056.PV durante operación

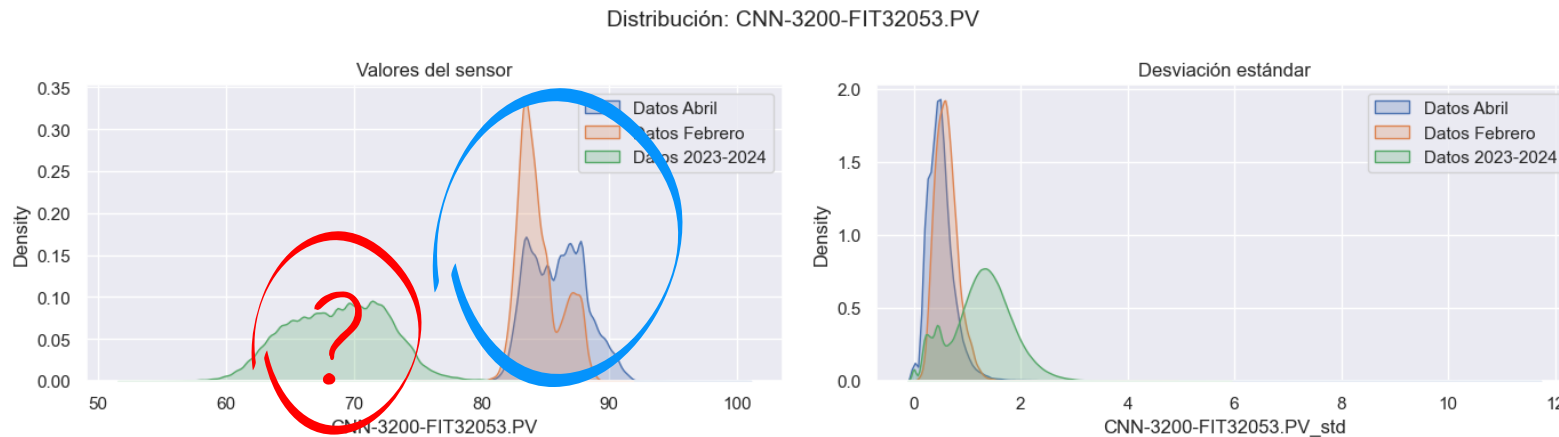


Selección de variables claves



Comportamiento bimodal en algunos sensores:

Al analizar la distribución de variables se detectaron algunos sensores que tenían una distribución bimodal de sus valores.

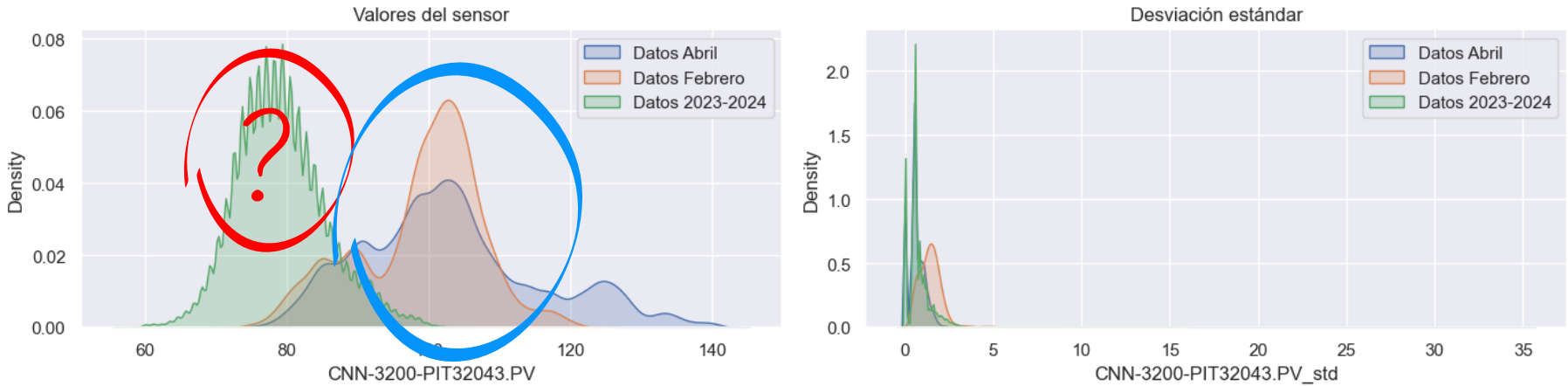


Comparación de comportamiento entre distintos periodos:

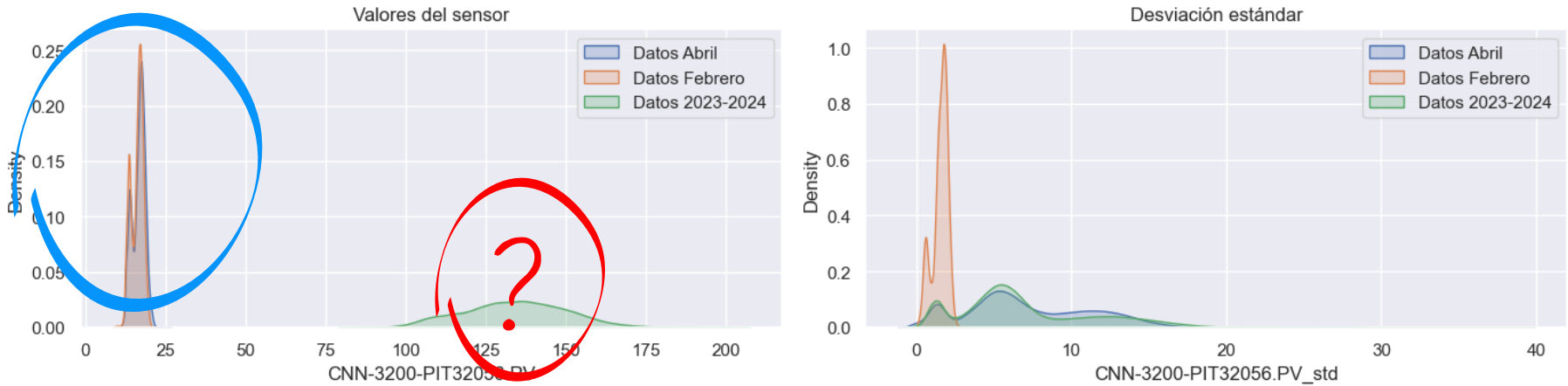
Al comparar la distribución de los datos históricos (2023-2024) con datos más actuales (Febrero-Abril 2025) se puede apreciar un cambio en el comportamiento normal de los valores.

Selección de variables claves

Distribución: CNN-3200-PIT32043.PV



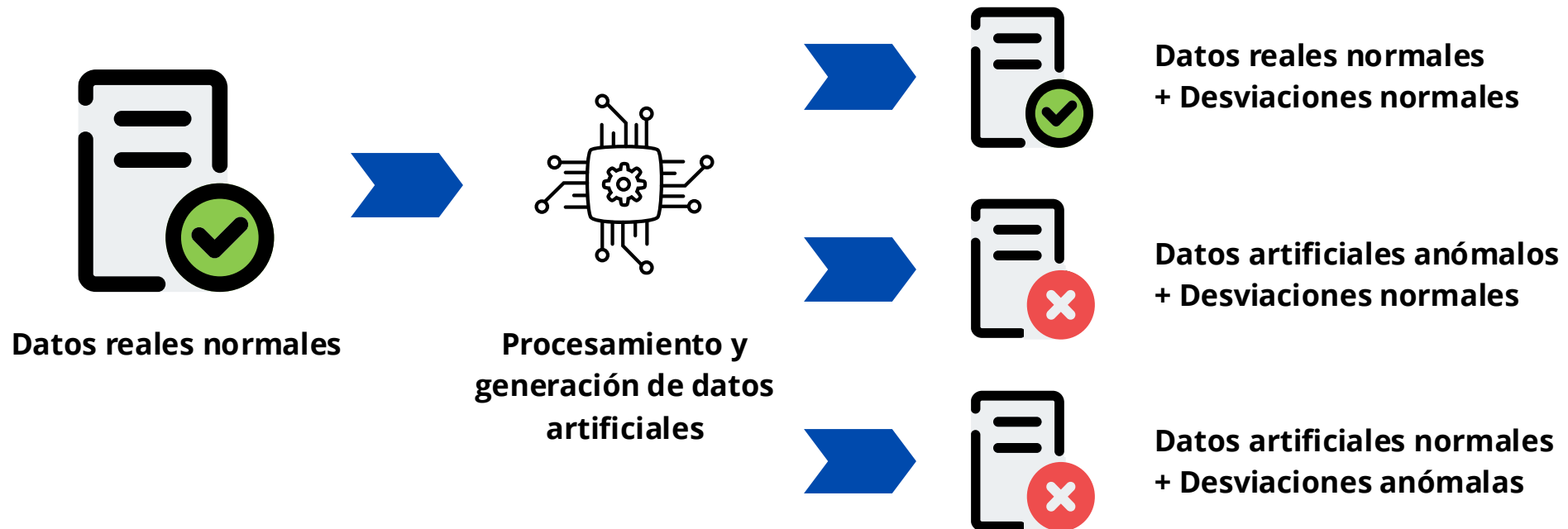
Distribución: CNN-3200-PIT32056.PV



Entrenamiento de los modelos

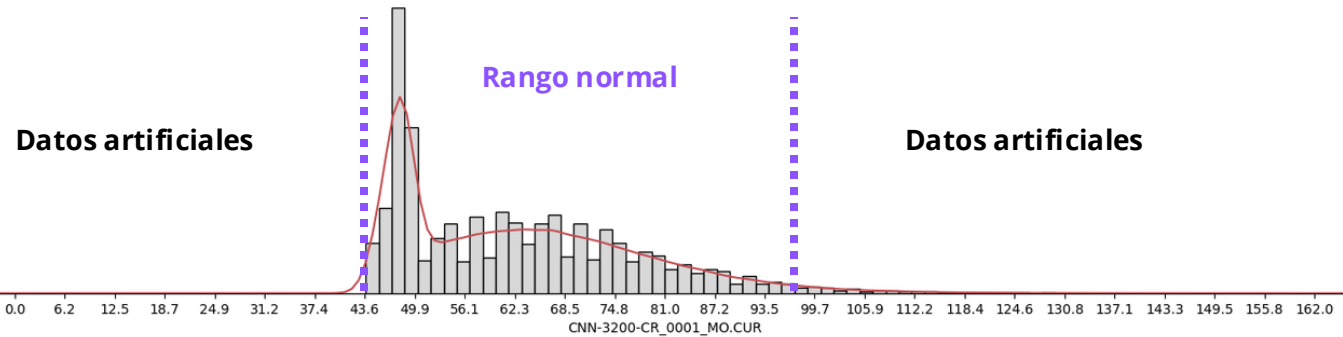
Conclusiones respecto a la selección de datos para el entrenamiento:

Dado el cambio de estado en los valores luego de los mantenimientos realizados en Diciembre 2024, se toma en consideración entrenar los modelos de Machine Learning con los datos actuales como "comportamiento normal" y generar datos artificiales que fueran anómalos según el comportamiento actual para considerarlos como estado de falla.

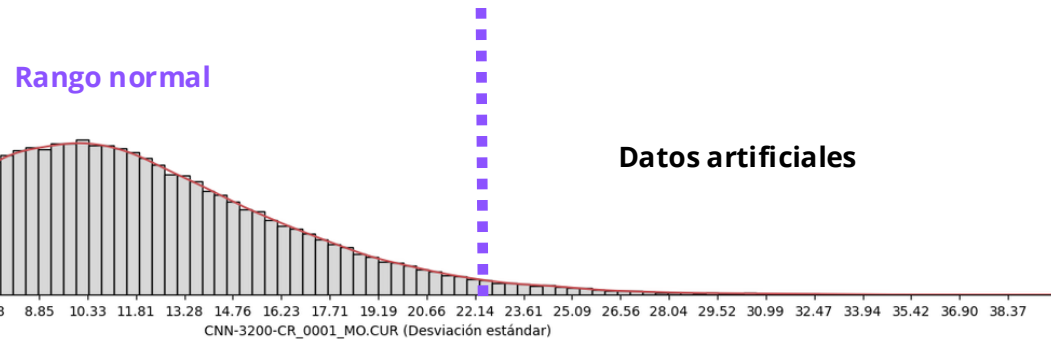


Entrenamiento de los modelos

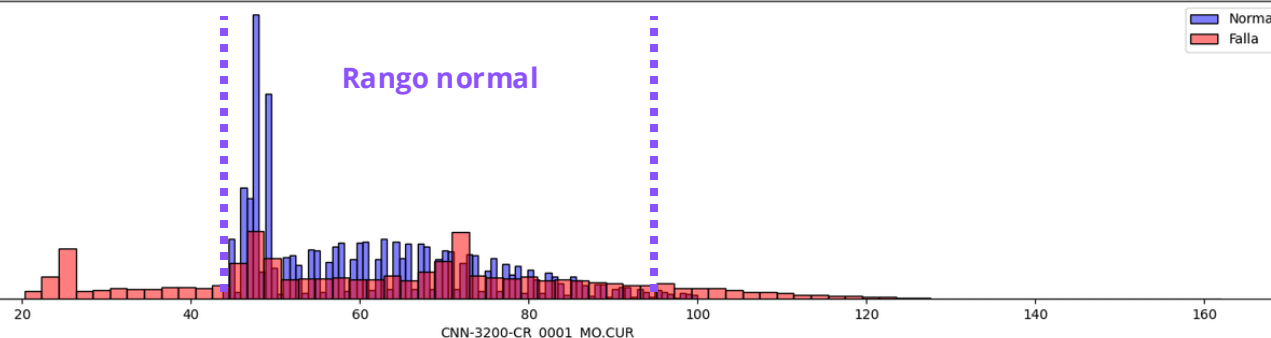
Distribución de CNN-3200-CR_0001_MO.CUR durante operación



Distribución de la desviación estándar de CNN-3200-CR_0001_MO.CUR en ventanas de 1 minuto



Comparación de distribución - CNN-3200-CR_0001_MO.CUR



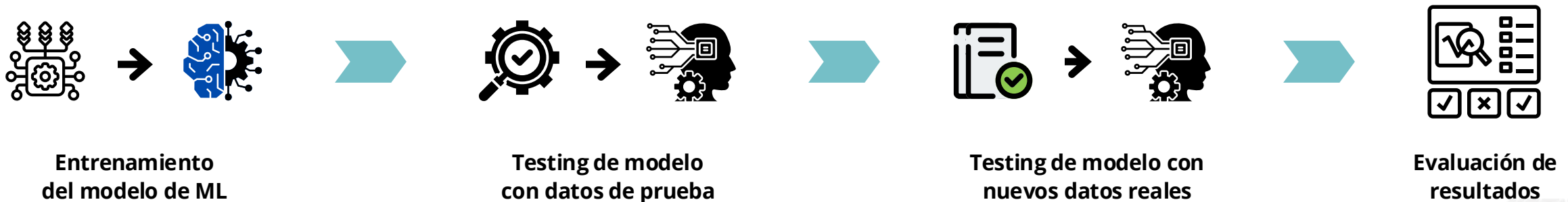
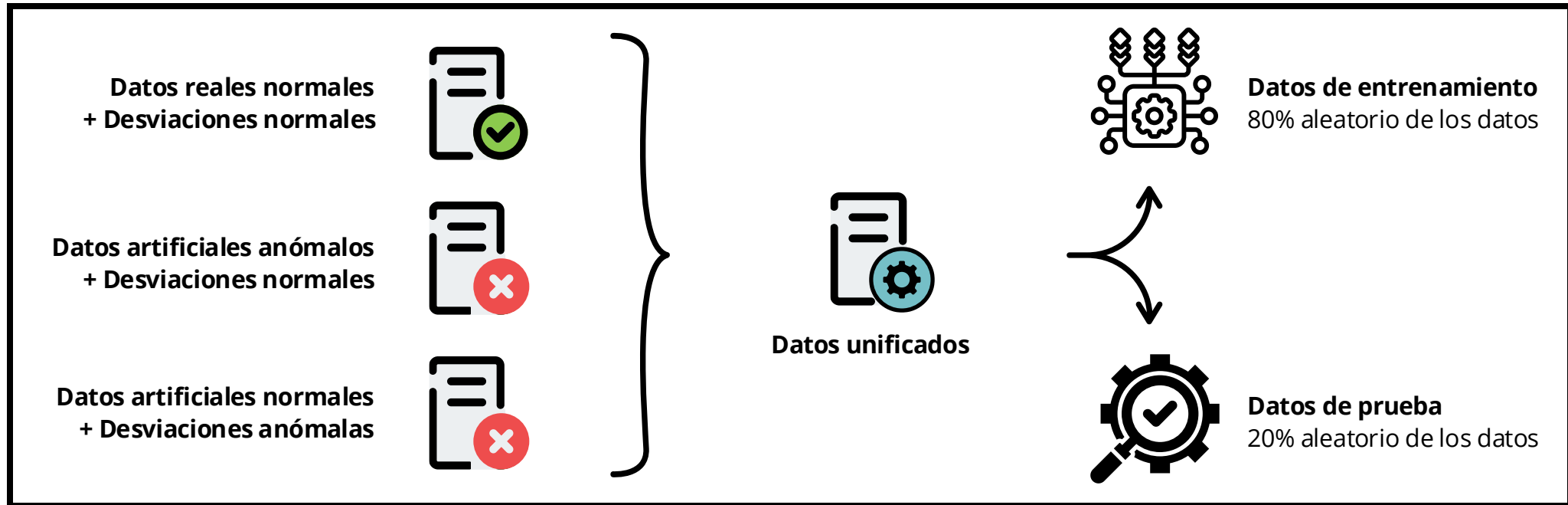
Definición de límites de operación normales para la generación de datos artificiales:

Utilizando las distribuciones de valores de cada sensor y sus desviaciones estándar, se definen límites de valor normal para que cualquier valor de operación fuera de esos rangos sea considerado anómalo por el modelo y así se puedan generar datos artificiales de falla fuera de estos márgenes.

Comparación de valores normales con artificiales:

Se pueden ver valores "Falla" que están en los rangos normales, esto es porque son valores normales con anomalías artificiales en sus desviaciones estándar

Entrenamiento de los modelos



Evaluación de los modelos

Resultados en la evaluación del desempeño de cada modelo desarrollado:

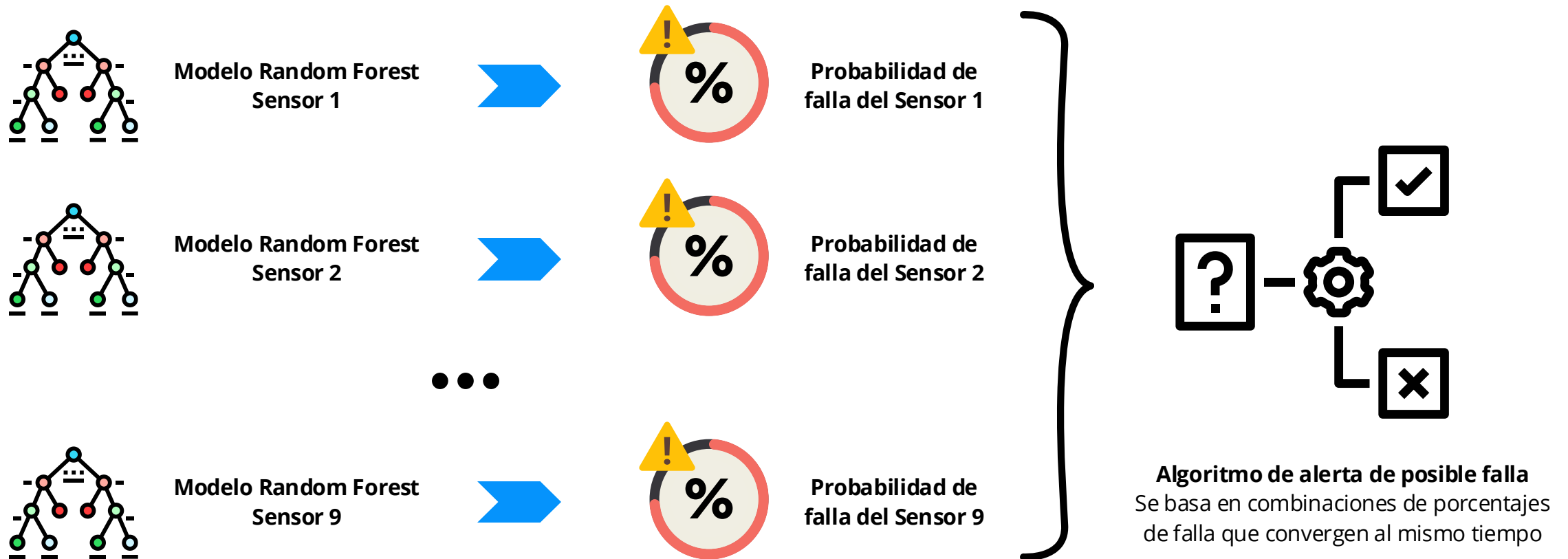
- **LSTM:** Si bien presenta muy buenos resultados en testing con datos de prueba, el modelo presenta falsos positivos con mucha frecuencia en testing con datos reales nuevos, mostrando claras señales de sobreajuste.
- **Autoencoders / Isolation Forest:** Modelos con alta tasa de falsos positivos en detección de fallas, debido a no contar con fallas reales suficientes para distinguir bien el comportamiento anómalo.
- **XGBoost / SVM:** Desempeño aceptable de los modelos, pero con mayor complejidad computacional y mucha sensibilidad al ruido de los datos.
- **Random Forest:** Mejor equilibrio entre rendimiento, interpretabilidad y velocidad. Alta precisión y fácil escalabilidad por sensor.



Modelo final adaptado

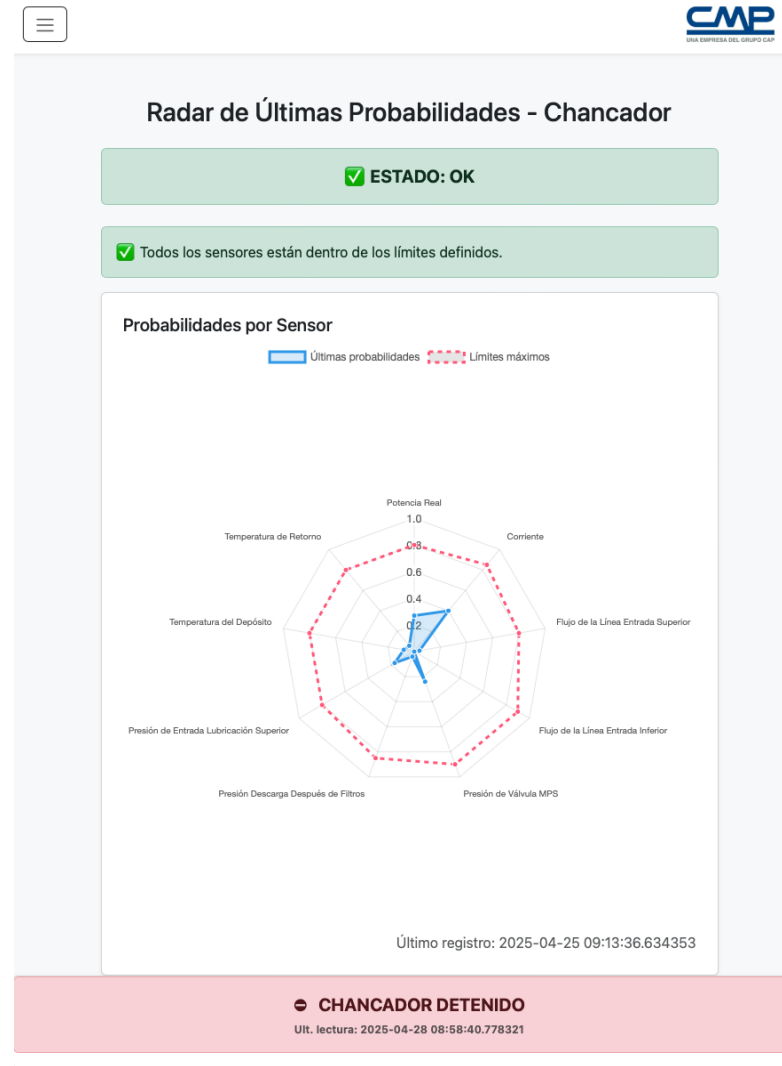
Lógica final desarrollada para el modelo con el desempeño más favorable:

Para aumentar el desempeño del modelo y mejorar sus capacidades de adaptabilidad a nuevas condiciones de funcionamiento en el chancador, se confeccionó un modelo de Random Forest Classifier independiente para cada uno de los sensores seleccionados. Con esto, se diseñó un algoritmo basado en las probabilidades de falla de cada sensor cuando en chancador se encuentra en funcionamiento activo.



Demo WebApp

Desarrollo de herramienta para la visualización para: Últimas predicciones; mostrar alertas.



Próximos pasos

- ▶ **Entrega de Machine learning en código fuente.**
- ▶ **Marcha blanca de la WebApp – 1 mes.**
- ▶ **Evaluar mantenimiento / soporte sistema M.L.**
- ▶ **Desarrollo escalable a sí mismo y a otros equipos.**

Conclusión – Impacto para CMP

Impacto para CMP en el uso de herramientas de Machine Learning

- ✓ Optimiza mantenimientos preventivos.
- ✓ Disminución riesgos operacionales.
- ✓ Mejora la disponibilidad de activos.
- ✓ Escalable a otros equipos críticos.