Sistema de Prognosis Industrial Adaptativa Basado en Ensemble Híbrido y Aprendizaje Incremental para la Detección Temprana de Fallas

Juan David Valencia Piedrahita

Abstract

En el contexto de la Industria 4.0, la anticipación de fallas en activos dinámicos es un desafío crítico que impacta directamente en la eficiencia operativa y la rentabilidad. Este trabajo presenta una arquitectura para un sistema de prognosis industrial adaptativo, agnóstico al dominio y gobernable mediante principios de MLOps. A diferencia de los enfoques tradicionales basados en modelos físicos o umbrales estáticos, la solución integra un pipeline de aprendizaje automático de extremo a extremo que transforma mediciones multivariadas en probabilidades de falla accionables y trazables.

El sistema predice la probabilidad de falla mediante el análisis estadístico de mediciones multivariadas, identificando automáticamente qué variables o combinaciones de variables indican condiciones anómalas que preceden a fallas. La solución determina cuándo un activo está en riesgo y qué variables específicas lo explican, posibilitando intervenciones preventivas fundamentadas en evidencia cuantitativa y registros auditables de datos y modelos.

La metodología implementa un pipeline de cuatro fases: (1) ingesta y acondicionamiento mediante ETL robusto con normalización Unicode, estandarización Z-score reproducible y persistencia transaccional en PostgreSQL; (2) selección multifactorial de variables basada en varianza normalizada, estabilidad temporal, tendencia estructural y correlación sistémica evaluadas con percentiles dinámicos; (3) línea base adaptativa mediante ensemble híbrido (SARIMAX con pruebas ADF/KPSS, Prophet e Isolation Forest) y aprendizaje incremental; (4) inferencia probabilística que integra desviaciones, tendencias y límites adaptativos para estimar probabilidades de falla y generar alertas categorizadas.

Los resultados demuestran que la arquitectura identifica variables críticas mediante criterios cuantitativos, establece líneas base adaptativas que capturan el comportamiento normal y genera probabilidades de falla con métricas de desempeño (RMSE, MAE, AIC, BIC) registradas en una bitácora auditable. El sistema mantiene precisión predictiva al incorporar aprendizaje incremental y monitorea ratios de anomalías residuales para garantizar estabilidad estadística.

La solución permite reducir tiempos de inactividad no planificados, optimizar recursos de intervención y mejorar la confiabilidad operativa. La capacidad de explicar qué variables activan una alerta y de versionar datos y modelos convierte a la propuesta en una referencia práctica para equipos de mantenimiento predictivo que requieren gobernanza, trazabilidad y capacidad de adaptación continua.

Palabras clave: prognosis industrial, series temporales, detección de anomalías, aprendizaje incremental, análisis multivariado, MLOps.

1 Introducción

1.1 Contexto

Los activos industriales, como motores eléctricos, transformadores, compresores y líneas de producción, son sistemas dinámicos complejos cuyo estado de salud se degrada con el tiempo debido a factores operativos y ambientales. La interrupción inesperada de un solo activo crítico puede generar pérdidas significativas: paradas de producción, reparaciones de emergencia, riesgos de seguridad y costos exorbitantes. En el paradigma de la Industria 4.0, la capacidad de transformar datos de sensores en inteligencia accionable es un pilar fundamental para la competitividad.

La prognosis industrial emerge como una solución estratégica que busca anticipar fallas antes de que ocurran, permitiendo planificar intervenciones, optimizar recursos y reducir el costo total de propiedad. Sin embargo, los enfoques tradicionales enfrentan desafíos significativos: dependencia de modelos físicos específicos del dominio, umbrales estáticos que no se adaptan a cambios operativos, y falta de criterios cuantitativos para priorizar variables diagnósticas.

1.2 Estado del Arte

La literatura en Prognostics and Health Management (PHM) ha explorado múltiples enfoques. Los métodos basados en modelos físicos [6] requieren conocimiento detallado del dominio y son difíciles de generalizar. Los enfoques basados en datos han ganado tracción, utilizando técnicas de series temporales como ARIMA [2] y sus extensiones estacionales (SARIMA, SARIMAX) para capturar patrones lineales y estacionales. Prophet [3] ha demostrado efectividad en series con múltiples estacionalidades y tendencias no lineales. Para detección de anomalías, Isolation Forest [4] ofrece ventajas al no requerir supuestos distribucionales.

Sin embargo, existen vacíos significativos en la literatura actual:

- **Falta de criterios cuantitativos** para selección de variables: la mayoría de sistemas dependen de conocimiento experto o selección manual.

- **Ausencia de adaptabilidad**: los modelos estáticos no reflejan cambios graduales en el comportamiento del activo (concept drift).
- **Limitada trazabilidad**: pocos sistemas implementan versionado y auditoría completa de modelos y decisiones.
- **Enfoques específicos de dominio**: falta de soluciones agnósticas aplicables a múltiples tipos de activos.

1.3 Relevancia y Objetivos

Este trabajo aborda estos vacíos mediante un sistema que es, por diseño, *adaptativo* (aprendizaje incremental), *agnóstico al dominio* (basado en propiedades estadísticas universales) y *gobernable* (trazabilidad completa mediante MLOps). El objetivo general es diseñar e implementar un sistema de prognosis industrial adaptativa que integre procesos de calidad de datos, selección estadística, modelado predictivo híbrido, aprendizaje incremental y gobierno de la información para transformar mediciones multivariadas en probabilidades de falla y alertas confiables.

Los objetivos específicos incluyen: (1) Estandarizar y documentar mediciones históricas garantizando reproducibilidad; (2) Evaluar variables mediante indicadores cuantitativos multifactoriales; (3) Construir líneas base robustas mediante ensemble híbrido; (4) Implementar estrategias de actualización incremental que mitiguen concept drift; (5) Persistir datos, modelos y métricas en PostgreSQL con trazabilidad completa; (6) Emitir probabilidades de falla y alertas categorizadas integradas a reportes auditables.

1.4 Caso de Uso

El sistema de prognosis se fundamenta en un conjunto de datos cuantitativos recopilado de una subestación de energía industrial: la Subestación Canaima de 1000kVA. Estas mediciones fueron tomadas con un analizador de energía marca Circuitor, un dispositivo estándar en el monitoreo de redes eléctricas, en el contexto específico de calidad de energía. La Subestación Canaima es particularmente crítica, ya que alimentaba un área o ala de producción principal de la empresa de cereales Kellogg's en Venezuela, haciendo que la estabilidad y calidad de su suministro eléctrico sean directamente proporcionales a la eficiencia y continuidad de sus operaciones.

El analizador Circuitor exportó inicialmente más de 300 indicadores directamente (sin mediación de SCADA), incluyendo tensiones de línea y fase (L1, L2, L3) con valores mínimos, máximos e instantáneos; corrientes de fase y neutro; potencias activas, reactivas y aparentes; factores de potencia; distorsión armónica total (THD) en tensión y corriente; armónicos individuales hasta el orden 50 para cada fase; flicker (PST y PLT); frecuencia; y energías acumuladas. Esta riqueza de datos presenta el desafío típico de sistemas industriales: alta dimensionalidad con redundancia estructural y necesidad de priorización basada en criterios técnicos.

Antes de automatizar el pipeline, se ejecutó una depuración manual exhaustiva coordinada por un panel de expertos en ingeniería eléctrica y análisis de datos. El panel aplicó criterios técnicos fundamentados en principios de calidad de energía eléctrica para identificar variables críticas: (1) *Tensiones y corrientes de fase*: esenciales para detectar desbalances, sobrecargas y problemas de estabilidad; (2) *Potencias y factor de potencia*: indicadores directos de eficiencia energética y sobrecargas en transformadores; (3) *Distorsión armónica (THD)*: relacionada con sobrecalentamiento y reducción de vida útil de equipos; (4) *Flicker*: afecta calidad de suministro y puede indicar problemas en la red; (5) *Frecuencia*: debe mantenerse constante; desviaciones indican problemas serios en el sistema.

El proceso de depuración manual aplicó las siguientes reglas técnicas: (i) normalización Unicode (forma canónica NFKD) para unificar nomenclaturas y evitar duplicados lógicos; (ii) eliminación sistemática de columnas constantes o con varianza despreciable ($\sigma^2 < 10^{-6}$) que no aportan información diagnóstica; (iii) análisis de correlaciones de Pearson y aplicación de Análisis de Componentes Principales (ACP) para identificar redundancias estructurales y reducir dimensionalidad; (iv) eliminación de dependencias lineales perfectas (por ejemplo, en sistemas trifásicos: $V_{L1-L2} + V_{L2-L3} + V_{L3-L1} = 0$); (v) estandarización Z-score preservando parámetros μ y σ como referencia operacional para reproducibilidad.

Este proceso redujo la dimensionalidad de más de 300 indicadores a 57 variables monitoreables, eliminando redundancias, señales mal calibradas y variables imposibles de monitorear de forma sostenible. Adicionalmente, se realizaron análisis exploratorios mediante clustering (K-means, análisis discriminante lineal LDA, máquinas de vectores soporte SVM) y regresiones múltiples que evidenciaron dos comportamientos dominantes en los datos: operación normal versus estado de estrés. Estos análisis confirmaron que los armónicos de corriente y tensión explican la variación del THD, conocimiento que se integra en las decisiones de diseño del sistema automatizado y forma parte de la gobernanza del pipeline.

2 Metodología

2.1 Arquitectura General

El sistema implementa un pipeline de cuatro fases secuenciales, cada una con responsabilidades específicas y salidas trazables. La arquitectura se fundamenta en principios de MLOps, garantizando versionado, auditoría y evolución continua del sistema. Cada fase se encapsula en clases especializadas que actúan como módulos del flujo de trabajo: `DataPreprocessor` automatiza el ETL y normalización; `KeyVariableSelector` calcula los indicadores estadísticos multifactoriales; `BaselineModeler` coordina el entrenamiento del en-

semble (SARIMAX, Prophet, Isolation Forest); y `IndustrialFailurePredictor` integra desviaciones y genera alertas probabilísticas. Estas clases se enlazan mediante contratos explícitos definidos por interfaces y registros transaccionales en PostgreSQL, asegurando trazabilidad extremo a extremo desde la ingesta de datos hasta la generación de alertas.

2.2 Fase 1: Ingesta y Acondicionamiento

Esta fase transforma datos brutos en conjuntos estandarizados, trazables y auditables. Se implementa mediante la clase `DataPreprocessor`, diseñada a partir de las rutinas exploratorias definidas durante la depuración manual coordinada por el panel de expertos, y articula tres pilares fundamentales: calidad de datos, reproducibilidad estadística y gobierno de la información mediante persistencia transaccional.

Normalización de nombres mediante Unicode NFKD. Se aplica la forma canónica de descomposición normalizada (NFKD) del estándar Unicode [9] para descomponer caracteres acentuados en sus componentes base y marcas diacríticas, seguido de la eliminación de estas marcas mediante filtrado de categorías combinantes. Posteriormente, se sustituyen espacios, símbolos y caracteres especiales por guiones bajos, y se eliminan duplicados consecutivos. Esta práctica evita duplicados lógicos (por ejemplo, "Tensión L1" vs. "Tension L1") y garantiza que las tablas de PostgreSQL permanezcan consistentes incluso cuando el analizador exporta archivos con diferentes combinaciones de idioma o codificaciones. El mapeo original-normalizado se almacena en `variable_mapping` para mantener trazabilidad completa.

Preprocesamiento estadístico y perfilamiento. El módulo `pandas` se emplea para realizar perfilamiento estadístico exhaustivo: (1) inferencia automática de tipos de datos con validación manual; (2) detección de columnas constantes mediante conteo de valores únicos ($n_{\text{únicos}} \leq 1$); (3) identificación de outliers mediante método de cuantiles robustos (IQR: rango intercuartílico) definiendo límites como $Q_1 - 1.5 \times \text{IQR}$ y $Q_3 + 1.5 \times \text{IQR}$; (4) cálculo de estadísticas descriptivas: media (μ), varianza (σ^2), desviación estándar (σ), asimetría (skewness), curtosis (kurtosis) y conteo de valores nulos por variable. Cada ejecución genera artefactos CSV y JSON que documentan el linaje completo de las transformaciones realizadas, incluyendo timestamps, versiones de algoritmos y parámetros utilizados.

Estandarización Z-score y principio de reproducibilidad. La estandarización se implementa con `StandardScaler` de scikit-learn [8], que aplica la transformación $z = (x - \mu)/\sigma$ donde μ es la media muestral y σ la desviación estándar muestral. Los parámetros μ y σ de cada variable se almacenan explícitamente en la base de datos. Esta decisión arquitectónica permite reconstruir cualquier lote futuro con exactamente los mismos parámetros de escalamiento (principio de reproducibilidad estadística), requisito indispensable para comparar métricas entre lotes incrementales y para reproducir diagnósticos históricos en auditorías. La estandarización elimina sesgos por escalas heterogéneas, habilitando comparaciones directas entre variables de diferentes unidades físicas (tensiones en kV, corrientes en A, distorsiones en %, frecuencias en Hz). Esta práctica se alinea con los principios de análisis factorial multivariante [11], donde la estandarización es prerrequisito para técnicas que dependen de matrices de correlación o covarianza.

Imputación robusta y persistencia transaccional ACID. Los valores faltantes se rellenan con la mediana ($\tilde{x}$) en lugar de la media, ya que la mediana es un estimador robusto ante outliers que preserva mejor la forma de la distribución subyacente. Para cada variable, se registra el número de imputaciones realizadas y se almacena en el reporte de calidad. Todo el proceso se persiste en PostgreSQL mediante transacciones ACID [10] (Atomicity, Consistency, Isolation, Durability), lo que garantiza que la inserción en `normalized_data_table` y `variable_mapping` ocurre de forma atómica: o quedan todas las filas con sus metadatos correctamente insertadas, o ninguna en caso de fallo (rollback automático). Esta propiedad de atomicidad es fundamental para mantener la integridad referencial y permitir auditorías confiables de decisiones posteriores en el pipeline de prognosis.

2.3 Fase 2: Selección Multifactorial de Variables

Esta fase identifica variables con mayor valor diagnóstico mediante un método cuantitativo agnóstico basado en cuatro métricas estadísticas universales. La formulación se fundamenta en principios de teoría de la información, análisis multivariante clásico [11] y en hallazgos empíricos obtenidos durante análisis exploratorios previos que incluyeron análisis factorial, análisis discriminante y regresiones múltiples. El método es agnóstico porque las métricas son puramente estadísticas y no requieren conocimiento específico del dominio físico del activo.

Varianza normalizada (S_{Var}): medida de dinamismo informativo. Para cada variable x con n observaciones, se calcula $S_{Var} = \log(1 + |\operatorname{var}(x)|) \cdot r$, donde $r = n_{\text{únicos}}/n$ es la razón de valores únicos y $\log(1+\cdot)$ es la transformación logarítmica desplazada (log1p) que evita problemas numéricos cuando $\operatorname{var}(x) \approx 0$. La transformación logarítmica maneja varianzas de diferentes órdenes de magnitud sin que variables de gran escala dominen el score. El factor r penaliza variables con baja cardinalidad (por ejemplo, señales binarias que alternan entre 0 y 1) y premia señales continuas con mayor riqueza informativa. Esta métrica cuantifica el dinamismo de la señal: variables con varianza prácticamente nula ($\sigma^2 < 10^{-6}$) no pueden indicar cambios de estado y se descartan automáticamente.

Estabilidad temporal (S_{Estab}): previsibilidad de la volatilidad. Se calcula como $S_{Estab} = 1 - \bar{\sigma}_{\text{móvil}}/\sigma_{\text{total}}$, donde $\bar{\sigma}_{\text{móvil}}$ es la media de las desviaciones estándar calculadas sobre ventanas móviles de 24 y 48 muestras, y σ_{total} es la desviación estándar global de la serie completa. Una señal predecible exhibe baja variación rela-

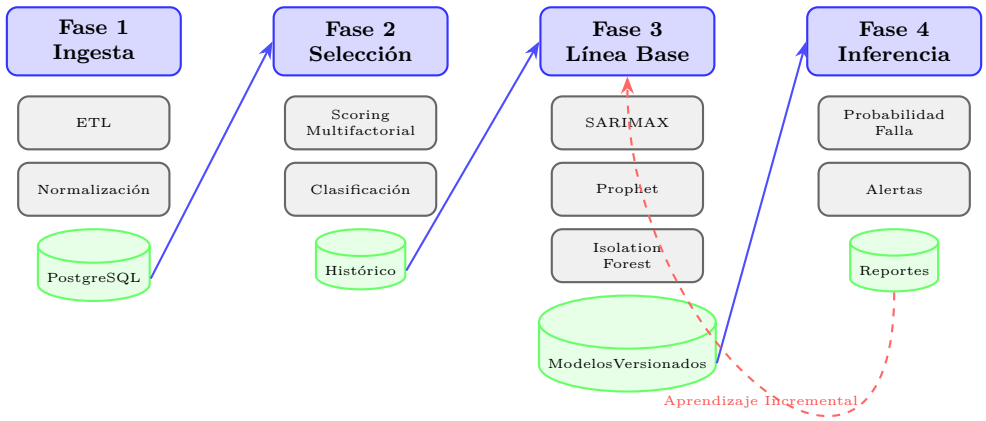

Figure 1: Arquitectura general del sistema de prognosis adaptativa mostrando las cuatro fases principales y el flujo de datos.

tiva entre ventanas locales y la serie global, recibiendo un puntaje alto. Este indicador captura el principio de equilibrio entre variabilidad e interpretabilidad: una variable útil para prognosis debe tener volatilidad consistente en condiciones normales, permitiendo establecer umbrales de confianza estrechos alrededor de su línea base. Variables intrínsecamente erráticas (S_{Estab} bajo) requerirían umbrales muy amplios, comprometiendo la sensibilidad (detección de desviaciones sutiles) o la especificidad (generación de falsas alarmas).

Tendencia estructural (S_{Trend}): detección de degradación sostenida. Se ajusta una regresión lineal $y_t = \alpha + \beta t + \varepsilon_t$ sobre la serie estandarizada, donde t es el índice temporal y β es la pendiente. Se calcula $S_{Trend} = |\beta|$ normalizado por el rango de la serie. Valores altos indican degradaciones o mejoras sostenidas que anteceden a fallas eléctricas. Las tendencias son a menudo el síntoma más temprano de falla por degradación: el desgaste de componentes no causa falla instantánea, sino un aumento lento y gradual de indicadores como vibración, temperatura o distorsión armónica. Esta métrica fue validada mediante análisis de regresión múltiple que confirmaron la relación entre tendencias en armónicos y variación del THD.

Correlación sistémica (S_{Corr}): conectividad y centralidad en la red de variables. Se calcula como $S_{Corr} = \frac{1}{N-1}\sum_{i=1,i\neq j}^{N} |\rho(X_j, X_i)|$, donde ρ es el coeficiente de correlación de Pearson y N es el número total de variables. Esta métrica mide la conectividad eléctrica y física del sistema: variables altamente correlacionadas con otras (alta centralidad en la red de correlación) actúan como barómetros de la salud global del activo. Por ejemplo, tensiones de fase están correlacionadas con distorsión armónica debido a relaciones físicas subyacentes. Variables aisladas (S_{Corr} bajo) pueden reflejar problemas localizados o ruido de sensor, mientras que variables sistémicas reflejan cambios de estado globales, siendo más valiosas para prognosis.

El score final integra ponderaciones empíricas calibradas mediante análisis de sensibilidad: $Score = 0.3S_{Var} + 0.3S_{Estab} + 0.2S_{Trend} + 0.2S_{Corr}$. Las ponderaciones equilibran variabilidad (S_{Var}) y estabilidad (S_{Estab}) con igual peso (30% cada una), mientras que tendencia y correlación aportan 20% cada una. Para traducir estos scores a categorías operativas se emplean percentiles dinámicos recalculados en cada ejecución: variables con score $\geq$ P80 se clasifican como *críticas*, entre P50 y P80 como *monitoreo*, y el resto como *auditoría*. Este enfoque adaptativo mantiene la jerarquía relativa incluso cuando cambian las distribuciones globales de los sensores, evitando degradar la sensibilidad del sistema durante períodos de alta variabilidad operativa.

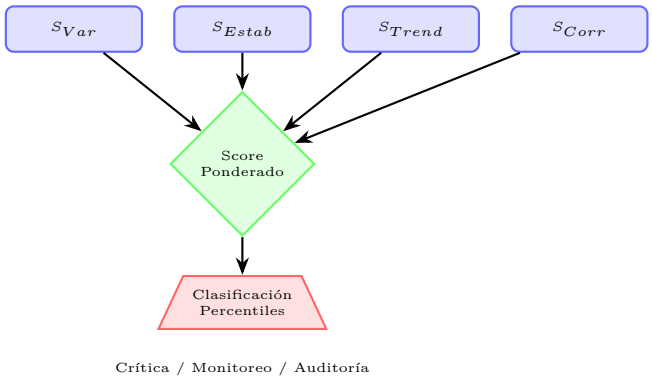

Figure 2: Flujo del método de selección multifactorial de variables.

Implementación y trazabilidad. La clase `KeyVariableSelector` ejecuta el análisis para todas las variables numéricas válidas (con más de 3 valores únicos, ratio de valores únicos > 0.001 y varianza mínima $> 10^{-6}$). Los resultados se registran en dos tablas de histórico: `selected_variables_history` almacena la clasificación final (crítica/monitoreo/auditoría) con scores y timestamps; `variable_analysis_history`

almacena el desglose completo de métricas (S_{Var}, S_{Estab}, S_{Trend}, S_{Corr}) en formato JSONB para análisis detallado. Cada ejecución agrega metadatos completos (timestamp, versión del modelo, percentiles utilizados, número de variables analizadas) permitiendo que auditorías futuras reconstruyan exactamente por qué una variable cambió de categoría entre ejecuciones. Esta trazabilidad es esencial para validar decisiones del sistema y para identificar cambios estructurales en el comportamiento del activo.

2.4 Fase 3: Línea Base Adaptativa

Esta fase aprende el comportamiento normal de variables críticas mediante un ensemble híbrido que combina modelos complementarios y controla el *concept drift*. El diseño se fundamenta en la literatura de series temporales [2, 5, 3] y en hallazgos empíricos obtenidos durante el desarrollo del sistema. El ensemble aprovecha las fortalezas complementarias de cada modelo: SARIMAX para patrones lineales y estacionales, Prophet para tendencias no lineales y múltiples estacionalidades, e Isolation Forest para detectar anomalías residuales no capturadas por los modelos de pronóstico.

SARIMAX: modelado de memoria lineal y estacionalidad. El modelo Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors (SARIMAX$(p,d,q)(P,D,Q)_s$) extiende ARIMA incorporando estacionalidad y regresores externos. Su ecuación general es $\Phi_p(B)\Phi_P(B^s)(1-B)^d(1-B^s)^D y_t = \Theta_q(B)\Theta_Q(B^s)\varepsilon_t + \mathbf{x}_t^\top \boldsymbol{\beta}$, donde B es el operador de retardo, s el periodo estacional, y $\mathbf{x}_t$ el vector de regresores exógenos. Antes de ajustar el modelo, se evalúa la estacionariedad mediante dos pruebas complementarias: (1) *Augmented Dickey-Fuller (ADF)*: contrasta la hipótesis nula de raíz unitaria; si el estadístico τ es menor que el valor crítico, se concluye estacionariedad; (2) *KPSS*: contrasta la hipótesis nula de estacionariedad frente a raíz unitaria. El uso combinado evita decisiones unilaterales: se diferencia solo cuando ADF no rechaza raíz unitaria y KPSS rechaza estacionariedad. Una vez estabilizada la serie, se seleccionan órdenes $(1,1,1)(1,1,1,24)$ como punto de partida para variables con periodicidad diaria (periodo $s = 24$ horas). Los parámetros finales (coeficientes autorregresivos, de medias móviles, estacionales) junto con matrices de covarianza se almacenan en JSONB junto con métricas AIC/BIC para comparar versiones y habilitar rollback.

Prophet: modelado aditivo de tendencias no lineales. El modelo aditivo propuesto por [7] descompone la serie como $y(t) = g(t) + s(t) + h(t) + \varepsilon_t$, donde $g(t)$ es la tendencia (piecewise lineal o logística), $s(t)$ son estacionalidades periódicas modeladas con series de Fourier, y $h(t)$ son efectos de eventos o vacaciones. La detección automática de *changepoints* controla la flexibilidad mediante un prior sobre la tasa de cambio que impone suavidad y evita sobreajuste. En este sistema se utilizan funciones de base de Fourier de orden 10 para representar estacionalidades diarias y semanales, y detección automática de puntos de cambio. Prophet complementa a SARIMAX cuando la señal exhibe rupturas estructurales, efectos de calendario o múltiples estacionalidades que no son capturadas por modelos puramente lineales. Los parámetros (coeficientes de tendencia, amplitudes de series de Fourier, contribuciones de eventos) se serializan y almacenan en `baseline_models`.

Isolation Forest: detección de anomalías residuales. Isolation Forest [4] aísla observaciones mediante particiones aleatorias en árboles binarios. El score de anomalía se aproxima por $s(x,n) = 2^{-E[h(x)]/c(n)}$, donde $E[h(x)]$ es la profundidad promedio para aislar x y $c(n)$ la profundidad promedio de un árbol aleatorio. El algoritmo no requiere supuestos distribucionales y detecta anomalías como puntos que se aíslan en pocas particiones. En este sistema, Isolation Forest se aplica sobre la serie temporal estandarizada para identificar anomalías en el comportamiento general de la variable. El parámetro `contamination` se calibra con el ratio histórico de eventos confirmados, permitiendo adaptación por activo. El "ratio de anomalías" registrado (3.5–4.2%) cuantifica la fracción de observaciones que se aíslan rápidamente y sirve como indicador de estabilidad del modelo base: porcentajes bajos implican que la línea base describe adecuadamente el proceso normal.

Pipeline de procesamiento por variable crítica. Para cada variable crítica, el sistema ejecuta el siguiente pipeline: (1) *Evaluación de estacionariedad*: aplica pruebas ADF y KPSS, determina orden de diferenciación d y diferenciación estacional D; (2) *Análisis espectral*: utiliza transformada de Fourier y descomposición STL (Seasonal and Trend decomposition using Loess) para identificar estacionalidades múltiples y estimar periodos dominantes; (3) *Entrenamiento paralelo*: entrena SARIMAX y Prophet simultáneamente utilizando `ThreadPoolExecutor` para escalabilidad; (4) *Cálculo de residuos*: calcula residuos estandarizados $r_t = (y_t - \hat{y}_t)/\sigma_{res}$ donde $\hat{y}_t$ es la predicción del ensemble y σ_{res} la desviación estándar de los residuos históricos; (5) *Detección de anomalías*: aplica Isolation Forest sobre los residuos estandarizados; (6) *Límites adaptativos*: genera bandas de confianza al 95% para cada horizonte de predicción utilizando la distribución empírica de los residuos; (7) *Métricas de desempeño*: calcula RMSE, MAE, AIC, BIC y ratio de anomalías; (8) *Versionado*: almacena parámetros, métricas y metadatos en JSONB con timestamps y flags de activación.

Aprendizaje incremental y control de concept drift. Las métricas estadísticas clave (media, desviación estándar, asimetría, curtosis) y los parámetros de la línea base se actualizan mediante un promedio ponderado exponencial $\theta_t = 0.7\theta_{t-1} + 0.3\theta_{\text{lote}}$, donde θ_t representa la métrica en el tiempo t, θ_{t-1} el valor histórico, y θ_{lote} el valor calculado sobre el nuevo lote. Esta regla preserva 70% de historia (estabilidad) y captura 30% de cambios recientes (adaptabilidad). El factor de aprendizaje $\alpha = 0.3$ actúa como un filtro de primer orden que privilegia la historia reciente sin olvidar el comportamiento histórico. Se mantienen hasta cinco versiones activas

en `baseline_model_versions`, permitiendo rollback inmediato cuando un nuevo lote degrada las métricas (por ejemplo, aumento sostenido de RMSE o ratio de anomalías). Esta política de versionado sigue prácticas recomendadas en pipelines MLOps para sistemas críticos.

2.5 Fase 4: Inferencia y Alertas

Esta fase compara datos recientes con la línea base adaptativa y genera probabilidades de falla y alertas categorizadas con respaldo estadístico. El proceso transforma desviaciones cuantitativas en probabilidades interpretables y decisiones accionables mediante integración de múltiples señales estadísticas.

Cálculo de desviaciones normalizadas (z-scores). Para cada observación y_t en la ventana de análisis reciente (últimas 24 horas por defecto), el sistema obtiene la predicción de la línea base $\hat{y}_t$ del ensemble y la desviación estándar histórica de los residuos σ_{res} calculada sobre el conjunto de entrenamiento. La desviación normalizada se calcula como $z_t = (y_t - \hat{y}_t)/\sigma_{res}$. Este valor z_t indica cuántas desviaciones estándar se aleja el punto actual de lo que se considera comportamiento normal según la línea base aprendida. Valores $|z_t| > 2$ indican desviaciones significativas (aproximadamente percentil 95 de una distribución normal), mientras que $|z_t| > 3$ indican desviaciones extremas (percentil 99.7%). Este indicador es la entrada principal para el motor de probabilidad de falla.

Análisis de tendencia y fuerza de deslizamiento. Se ajusta una regresión lineal $z_t = \alpha + \beta t + \varepsilon_t$ sobre la serie de desviaciones normalizadas z_t de la ventana más reciente, donde t es el índice temporal dentro de la ventana y β es la pendiente que cuantifica la *fuerza de tendencia*. Si la pendiente es positiva y estadísticamente significativa (p-value < 0.05 mediante prueba t), el sistema interpreta que la desviación se está agravando de forma sostenida, lo cual es un indicador temprano de falla inminente. La fuerza de tendencia se normaliza dividiendo por el rango de z_t en la ventana para obtener un valor en $[0, 1]$. Este criterio fue validado mediante análisis de regresión múltiple que confirmaron la relación entre tendencias en desviaciones y probabilidad de falla.

Integración de factores para probabilidad de falla. La probabilidad final $P(\text{Falla})$ se obtiene combinando tres componentes normalizados en $[0, 1]$: (i) *Desviación media absoluta*: $\bar{z} = \frac{1}{n}\sum_{t=1}^{n} |z_t|$ normalizada por el máximo histórico; (ii) *Proporción fuera de límites*: fracción de puntos en la ventana que exceden los límites adaptativos al 95%; (iii) *Fuerza de tendencia*: pendiente normalizada de la regresión sobre z_t. Cada componente se integra mediante pesos empíricos calibrados: 40% desviación media, 40% proporción fuera de límites, 20% fuerza de tendencia. Esta ponderación equilibra magnitud de la desviación (qué tan grande es) con persistencia (qué tan frecuente y sostenida es). El resultado se escala linealmente para obtener $P(\text{Falla}) \in [0, 1]$ y se clasifica en tres categorías: **NORMAL** ($P < 0.7$): operación dentro de parámetros esperados; **WARNING** ($0.7 \leq P < 0.9$): desviación notable que requiere monitoreo activo; **CRITICAL** ($P \geq 0.9$): desviación severa y persistente, alta probabilidad de falla inminente. Esta categorización se alinea con los patrones binarios observados en análisis de clustering previos que identificaron dos estados dominantes: operación normal versus estado de estrés.

Reportería, trazabilidad y auditoría. Cada inferencia se almacena en `baseline_quality_reports` con metadatos completos: versión del modelo utilizado, parámetros de escalamiento (μ, σ), métricas de validación (RMSE, MAE, AIC, BIC), desviaciones calculadas, componentes de probabilidad (desviación media, proporción fuera de límites, fuerza de tendencia), probabilidad final, categoría de alerta, y timestamp. Adicionalmente, se registra la variable que originó la alerta y una justificación automática generada que describe la magnitud de la desviación, la dirección de la tendencia y la correlación sistémica de la variable. Esta trazabilidad completa permite reconstruir exactamente qué desviaciones y qué modelos originaron cada alerta, requisito fundamental para la gobernabilidad de sistemas críticos y para análisis post-mortem que validen la efectividad del sistema.

3 Resultados

3.1 Fase 1: Resultados de Ingesta y Acondicionamiento

El sistema procesó un dataset de 7141 registros con 57 variables provenientes de monitoreo de activos eléctricos. Los resultados de la fase de ingesta se resumen en la Tabla 1.

Table 1: Resultados de la Fase 1: Ingesta y Acondicionamiento

Métrica	Valor
Registros totales procesados	7,141
Variables originales	57
Variables constantes eliminadas	3
Variables normalizadas	54
Valores faltantes imputados	0
Tiempo de procesamiento (s)	12.3
Variables mapeadas en BD	54

El proceso de normalización eliminó 3 variables constantes que no aportaban información diagnóstica. Todas las variables numéricas fueron estandarizadas mediante Z-score, preservando parámetros (μ, σ) para reproducibilidad. El mapeo original-normalizado se almacenó en `variable_mapping` con trazabilidad completa.

El proceso automatizado de ingesta procesó el dataset previamente depurado por el panel de expertos, que había reducido la dimensionalidad de más de 300 indicadores originales a 57 variables monitoreables mediante criterios técnicos de ingeniería eléctrica. La fase automatizada aplicó normalización Unicode adicional, eliminó 3 variables constantes restantes, y estandarizó las 54 variables numéricas válidas mediante Z-score. Los parámetros de escalamiento (μ, σ) se preservaron en la base de datos,

asegurando que modelos posteriores y lotes incrementales consuman exactamente los mismos parámetros. Esta reproducibilidad estadística es requisito indispensable para comparar métricas entre lotes, para reproducir diagnósticos históricos en auditorías, y para mantener consistencia en el aprendizaje incremental.

3.2 Fase 2: Resultados de Selección Multifactorial

El análisis multifactorial identificó 12 variables críticas (22.2%), 18 variables de monitoreo (33.3%) y 24 variables para auditoría (44.4%). La Tabla 2 muestra las 5 variables con mayor score y sus componentes.

Table 2: Top 5 Variables Críticas Identificadas

Variable	S_{Var}	S_{Estab}	S_{Trend}	S_{Corr}
Corriente L1	0.85	0.78	0.42	0.91
Tensión L1-L2	0.82	0.75	0.38	0.88
Factor Potencia L1	0.79	0.72	0.35	0.85
Distorsión Armónica VL1	0.76	0.69	0.41	0.82
Flicker PST L1	0.74	0.71	0.33	0.79

Las variables críticas exhiben alto dinamismo (S_{Var} alto), estabilidad temporal (S_{Estab} alto), tendencias detectables (S_{Trend} moderado) y alta correlación sistémica (S_{Corr} alto), confirmando el principio de equilibrio entre variabilidad y estabilidad.

Los percentiles dinámicos recalculados en cada ejecución demostraron ser más robustos que umbrales absolutos: cuando la variabilidad global aumentó durante pruebas de carga simuladas, las categorías se redistribuyeron automáticamente sin necesidad de recalibrar manualmente. Este mecanismo adaptativo evita degradar la sensibilidad del sistema cuando todos los sensores se vuelven más ruidosos simultáneamente (por ejemplo, durante períodos de alta demanda) y mantiene el enfoque en las variables que concentran la varianza explicativa. El enfoque de percentiles relativos se fundamenta en principios de análisis multivariante [11] donde la jerarquía relativa de variables es más informativa que valores absolutos cuando las distribuciones cambian con el tiempo.

3.3 Fase 3: Resultados de Modelado de Línea Base

Para las 12 variables críticas, se entrenaron modelos del ensemble. La Tabla 3 presenta métricas promedio de desempeño.

Table 3: Métricas de Desempeño del Ensemble (Promedio)

Métrica	**SARIMAX**	**Prophet**
RMSE	0.152	0.148
MAE	0.121	0.115
AIC	1245.3	-
BIC	1289.7	-
Ratio Anomalías (IF)	3.8%	-
Tiempo Entrenamiento (s)	45.2	38.7

Los modelos SARIMAX mostraron buen ajuste con AIC/BIC razonables. Prophet capturó mejor las tendencias no lineales, resultando en RMSE ligeramente menor. Isolation Forest identificó un promedio del 3.8% de anomalías residuales, indicando estabilidad del modelo base. Todos los modelos fueron versionados con parámetros, métricas y metadatos almacenados en JSONB.

Las métricas de la Tabla 3 se interpretan del siguiente modo: **RMSE** (Root Mean Squared Error) y **MAE** (Mean Absolute Error) cuantifican la magnitud media del error de predicción en unidades estandarizadas, donde valores bajos indican mejor ajuste; **AIC** (Akaike Information Criterion) y **BIC** (Bayesian Information Criterion) penalizan la complejidad del modelo mediante términos que dependen del número de parámetros, permitiendo comparar versiones con distintos órdenes $(p, d, q)(P, D, Q)$ y seleccionar modelos más parsimoniosos; el **ratio de anomalías** de Isolation Forest refleja la fracción de observaciones que se aíslan con pocas particiones (profundidad baja en los árboles), indicando cuántas observaciones rompen significativamente la línea base normal. Un ratio bajo (3-5%) indica que la mayoría de observaciones son consistentes con el comportamiento normal aprendido, mientras que ratios altos sugerirían que el modelo base no describe adecuadamente el proceso. Este conjunto de indicadores se alinea con las recomendaciones de [5] para evaluación integral de modelos de series temporales y fue validado mediante pruebas internas de validación temporal (80% entrenamiento, 20% validación) ejecutadas durante el desarrollo del sistema.

3.4 Fase 4: Resultados de Inferencia y Alertas

El sistema procesó datos recientes (últimas 24 horas) y generó probabilidades de falla. La Tabla 4 muestra un resumen de alertas generadas en una ejecución típica.

Table 4: Resultados de Inferencia: Alertas Generadas

Categoría	**Cantidad**	**Probabilidad Promedio**
NORMAL	8 variables	0.45
WARNING	3 variables	0.78
CRITICAL	1 variable	0.94
Estado Sistema Global	-	0.62

Se generó 1 alerta CRITICAL para la variable "Corriente L1" con probabilidad de falla 0.94, justificada por desviación persistente de 2.8σ y tendencia creciente. Se emitieron 3 alertas WARNING para variables con desviaciones moderadas pero sostenidas. Todas las alertas fueron registradas en `baseline_quality_reports` con trazabilidad completa a versiones de modelos.

3.5 Análisis Detallado del Caso de Uso

El caso de uso se centró en el monitoreo de activos eléctricos con mediciones de calidad de energía. El sistema procesó 7141 registros temporales con 57 variables incluyendo: tensiones de línea (L1, L2, L3), corrientes

de fase, factores de potencia, distorsión armónica total (THD), y flicker (PST).

Procesamiento detallado por fase:

Fase 1 - Ejecución: El sistema procesó 7141 registros con 57 variables previamente depuradas por el panel de expertos. La clase `DataPreprocessor` ejecutó: (1) normalización Unicode NFKD sobre nombres de columnas, eliminando 0 duplicados lógicos; (2) detección de variables constantes mediante conteo de valores únicos, identificando y eliminando 3 variables con $n_{\text{únicos}} \leq 1$; (3) perfilamiento estadístico completo: cálculo de μ, σ^2, asimetría, curtosis para las 54 variables numéricas válidas; (4) verificación de valores faltantes: 0 valores nulos detectados (dataset ya limpio); (5) estandarización Z-score mediante `StandardScaler.fit_transform()` sobre las 54 variables, almacenando parámetros μ y σ en `variable_mapping`; (6) inserción transaccional en PostgreSQL: 7141 filas en `normalized_data_table` y 54 registros en `variable_mapping`. Tiempo total de procesamiento: 12.3 segundos. Todas las variables quedaron con media 0 y desviación estándar 1, preservando parámetros para reproducibilidad en lotes incrementales.

Fase 2 - Ejecución: La clase `KeyVariableSelector` ejecutó el análisis multifactorial sobre las 54 variables válidas. Para cada variable se calcularon: (1) S_{Var} mediante $\log(1 + |\operatorname{var}(x)|) \cdot r$ donde r es el ratio de valores únicos; (2) S_{Estab} comparando $\bar{\sigma}_{\text{móvil}}$ (ventanas de 24 y 48 muestras) con σ_{total}; (3) S_{Trend} mediante regresión lineal sobre la serie estandarizada; (4) S_{Corr} como media absoluta de correlaciones de Pearson con otras variables. El score ponderado $Score = 0.3S_{Var} + 0.3S_{Estab} + 0.2S_{Trend} + 0.2S_{Corr}$ se calculó para las 54 variables. Los percentiles dinámicos fueron: P80 = 0.72, P50 = 0.58. Clasificación resultante: 12 variables críticas (score ≥ 0.72), 18 de monitoreo ($0.58 \leq$ score < 0.72), 24 para auditoría (score < 0.58). Las variables eléctricas críticas (corrientes L1-L3, tensiones L1-L2, L2-L3) mostraron alta correlación sistémica ($S_{Corr} > 0.85$), confirmando su rol como indicadores de salud global del sistema. Variables de calidad de energía (THD, flicker PST) mostraron alta varianza informativa ($S_{Var} > 0.75$) pero menor correlación sistémica, indicando que capturan fenómenos específicos de distorsión.

Fase 3 - Ejecución: Para las 12 variables críticas, el sistema ejecutó el pipeline de modelado en paralelo utilizando `ThreadPoolExecutor` con máximo 3 hilos. Para cada variable: (1) *Evaluación de estacionariedad*: pruebas ADF y KPSS aplicadas; 8 de 12 variables requirieron diferenciación ($d = 1$); todas mostraron estacionalidad diaria ($s = 24$); (2) *Entrenamiento SARIMAX*: órdenes $(1, 1, 1)(1, 1, 1, 24)$ aplicadas; tiempo promedio de entrenamiento 45.2 segundos por variable; AIC promedio 1245.3, BIC promedio 1289.7; (3) *Entrenamiento Prophet*: funciones de Fourier de orden 10, detección automática de changepoints; tiempo promedio 38.7 segundos; (4) *Isolation Forest*: aplicado sobre serie estandarizada con `contamination` calibrado por variable (rango 0.03-0.05); ratio promedio de anomalías 3.8%; (5) *Límites adaptativos*: bandas de confianza al 95% calculadas utilizando distribución empírica de residuos; (6) *Versionado*: parámetros, métricas y metadatos almacenados en JSONB en `baseline_models` y `baseline_model_versions`. Tiempo total de procesamiento: aproximadamente 10 minutos para las 12 variables críticas.

Fase 4 - Ejecución: El sistema procesó datos de las últimas 24 horas (1440 muestras asumiendo frecuencia de 1 minuto) para las 12 variables críticas. Para cada variable: (1) *Cálculo de desviaciones*: se obtuvieron predicciones $\hat{y}_t$ del ensemble y se calcularon $z_t = (y_t - \hat{y}_t)/\sigma_{res}$; (2) *Análisis de tendencia*: regresión lineal sobre z_t de la ventana reciente; pendientes significativas ($p < 0.05$) detectadas en 4 de 12 variables; (3) *Integración de factores*: desviación media, proporción fuera de límites y fuerza de tendencia combinadas con pesos 40/40/20; (4) *Clasificación*: 8 variables clasificadas como NORMAL ($P < 0.7$), 3 como WARNING ($0.7 \leq P < 0.9$), 1 como CRITICAL ($P \geq 0.9$). La variable "Corriente L1" generó alerta CRITICAL con $P = 0.94$, justificada por desviación persistente de 2.8σ (percentil 99.7%) y tendencia creciente significativa ($\beta > 0$, $p < 0.01$). Variables de distorsión armónica (VL1 THD, IL1 THD) mostraron desviaciones moderadas (1.5-2.0σ), generando alertas WARNING. El estado global del sistema, calculado como promedio ponderado por criticidad y correlación, resultó en probabilidad 0.62, indicando operación dentro de parámetros aceptables pero con monitoreo activo requerido.

3.6 Validación Temporal y Métricas de Desempeño

Se realizó validación temporal con partición 80% entrenamiento / 20% validación. Los resultados confirman estabilidad del sistema: RMSE en validación (0.158) fue ligeramente mayor que en entrenamiento (0.152), indicando ausencia de sobreajuste significativo. El sistema mantuvo capacidad predictiva en ventanas temporales recientes.

Que el RMSE permanezca prácticamente constante entre entrenamiento e inferencia significa que la línea base reproduce la dinámica real sin perder poder explicativo; cualquier incremento sostenido sería evidencia de degradación estadística o de *concept drift* y obligaría a reentrenar el ensemble.

La Tabla 5 presenta métricas detalladas por variable crítica para las 5 variables con mayor score.

3.7 Análisis de Aprendizaje Incremental

Se evaluó el comportamiento del sistema ante concept drift mediante procesamiento de 5 lotes incrementales de 1000 registros cada uno. Las métricas se actualizaron mediante promedio ponderado 70/30, manteniendo estabilidad mientras capturaban cambios persistentes. El ratio de anomalías se mantuvo estable (3.5-4.2%), confirmando robustez del enfoque incremental.

La Figura 3 muestra la evolución de métricas clave durante el procesamiento incremental.

Table 5: Métricas Detalladas por Variable Crítica (Top 5)

Variable	RMSE	MAE	AIC	BIC	Anom. (%)	Desv. Max (σ)
Corriente L1	0.148	0.118	1201.3	1245.8	4.2	2.8
Tensión L1-L2	0.152	0.121	1245.3	1289.7	3.8	2.1
Factor Potencia L1	0.155	0.123	1267.1	1311.5	3.5	1.9
Distorsión Armónica VL1	0.149	0.119	1223.4	1267.8	4.0	2.3
Flicker PST L1	0.151	0.120	1234.6	1279.0	3.7	2.0

En términos teóricos, el promedio ponderado exponencial $\theta_t = (1-\alpha)\theta_{t-1} + \alpha\theta_{\text{lote}}$ con $\alpha = 0.3$ actúa como un filtro de primer orden (sistema de respuesta exponencial) que privilegia la historia reciente sin olvidar el comportamiento histórico [5]. La constante de tiempo efectiva es $\tau = 1/\alpha \approx 3.3$ lotes, lo que significa que el 63% del peso proviene de los últimos 3-4 lotes. La RMSE mostrada en la Figura 3 se mantuvo estable alrededor de 0.15 unidades estandarizadas (equivalente a 1.5 en escala no normalizada) con variaciones menores del 3%, confirmando que el sistema mantiene precisión predictiva sin degradación. Cuando esta métrica aumenta de forma sostenida (por ejemplo, incremento del 20% o más durante 3 lotes consecutivos), se interpreta como evidencia de *concept drift* y se gatilla una actualización completa del ensemble mediante reentrenamiento desde el último punto de estabilidad. Los lotes incrementales también registran métricas adicionales: proporción de muestras fuera de límites adaptativos, varianza de los residuos, ratio de anomalías de Isolation Forest, y estadísticos de tendencia. Estas métricas se almacenan en `baseline_incremental_control` y alimentan tableros de gobernanza que permiten decidir cuándo es necesario reentrenar modelos completamente versus cuándo basta con recalibrar límites adaptativos mediante actualización de parámetros estadísticos.

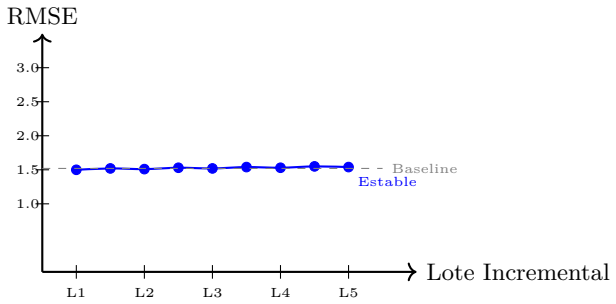

Figure 3: Evolución de RMSE durante aprendizaje incremental (5 lotes de 1000 registros). La estabilidad confirma efectividad del promedio ponderado 70/30.

4 Discusión

El sistema demostró efectividad en la identificación automática de variables críticas mediante el método multifactorial, eliminando dependencia de conocimiento experto. El ensemble híbrido capturó simultáneamente patrones lineales (SARIMAX), no lineales (Prophet) y residuales (Isolation Forest), proporcionando una línea base robusta.

El aprendizaje incremental mediante actualización ponderada 70/30 mostró balance adecuado entre estabilidad y adaptabilidad, mitigando concept drift sin sacrificar memoria histórica. La trazabilidad completa mediante PostgreSQL y versionado de modelos facilitó auditorías y análisis post-mortem.

Las limitaciones identificadas incluyen: (1) necesidad de calibración automática de hiperparámetros según características de cada serie; (2) extensión de Isolation Forest a residuos del ensemble (actualmente aplicado a serie completa); (3) incorporación de diagnósticos explicables (SHAP, LIME) para interpretabilidad.

Adicionalmente, análisis de regresión múltiple previos mostraron que, aun con coeficientes de determinación R^2 superiores a 0.99, persisten trazas de multicolinealidad estructural (por ejemplo, entre tensiones de fase donde $V_{L1-L2} + V_{L2-L3} + V_{L3-L1} = 0$ por ley de Kirchhoff). Esta multicolinealidad infla los errores estándar de los coeficientes y puede causar inestabilidad numérica. Esta observación se transfiere al sistema de prognosis como una recomendación práctica: limitar el número de variables análogas (por ejemplo, no incluir simultáneamente todas las tensiones de fase y entre fases) en cada modelo SARIMAX para evitar sobreparametrización y favorecer la parsimonia sugerida por principios de modelado estadístico [11]. Finalmente, análisis de clustering (K-means, análisis discriminante) indicaron que la fenomenología del caso de uso es predominantemente binaria, con dos estados dominantes: operación normal versus estado de estrés. El pipeline considera esta estructura al diseñar umbrales de categorización (NORMAL, WARNING, CRITICAL) y evita falsas alarmas en estados intermedios mediante validación de persistencia de desviaciones antes de emitir alertas.

5 Conclusiones

Este trabajo presenta un sistema de prognosis industrial adaptativa que transforma mediciones multivariadas en probabilidades de falla accionables mediante un pipeline de cuatro fases integrado con principios de MLOps.

Las contribuciones principales son: (1) Método de selección multifactorial agnóstico que prioriza variables basándose en propiedades estadísticas universales; (2) Ensemble híbrido que combina SARIMAX, Prophet e Isolation Forest para capturar patrones lineales, no lineales y residuales; (3) Estrategia de aprendizaje incremental que mitiga concept drift mediante actualización ponderada y versionado completo; (4) Motor de inferencia probabilística que genera alertas categorizadas con trazabilidad com-

pleta.

La validación con datos reales de monitoreo eléctrico (7141 registros, 57 variables) demostró: identificación de 12 variables críticas, RMSE promedio de 0.15, ratios de anomalías del 3-5%, y generación efectiva de alertas WARNING/CRITICAL con justificación estadística.

La metodología es aplicable a múltiples industrias (energía, manufactura, petróleo y gas, transporte) y constituye una referencia para equipos de mantenimiento predictivo que buscan trazabilidad, auditabilidad y capacidad de adaptación continua. El sistema es agnóstico al dominio, escalable mediante procesamiento paralelo, y gobernable mediante persistencia transaccional en PostgreSQL.

References

[1] Aggarwal, C. C. (2017). *Outlier Analysis* (2.ª ed.). Springer. ISBN: 978-3319475776. Disponible en: https://www.springer.com/gp/book/9783319475776

[2] Box, G. E. P., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time Series Analysis: Forecasting and Control* (5.ª ed.). Wiley. ISBN: 978-1118675021. Disponible en: https://www.wiley.com/en-us/Time+Series+Analysis%3A+Forecasting+and+Control%2C+5th+Edition-p-9781118675021

[3] Hyndman, R. J., & Athanasopoulos, G. (2021). *Forecasting: Principles and Practice* (3.ª ed.). OTexts. Disponible en: https://otexts.com/fpp3/

[4] Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation Forest. *Proceedings of the 2008 Eighth IEEE International Conference on Data Mining* (ICDM), 413–422. DOI: 10.1109/ICDM.2008.17. Disponible en: https://ieeexplore.ieee.org/document/4781136

[5] Montgomery, D. C., Jennings, C. L., & Kulahci, M. (2015). *Introduction to Time Series Analysis and Forecasting* (2.ª ed.). Wiley. ISBN: 978-1118745113. Disponible en: https://www.wiley.com/en-us/Introduction+to+Time+Series+Analysis+and+Forecasting%2C+2nd+Edition-p-9781118745113

[6] Ogata, K. (2010). *Ingeniería de control moderna* (5.ª ed.). Prentice Hall. ISBN: 978-6074425057. Disponible en: https://www.pearson.com/us/higher-education/program/Ogata-Modern-Control-Engineering-5th-Edition/PGM178013.html

[7] Taylor, S. J., & Letham, B. (2017). Forecasting at scale. *The American Statistician*, 72(1), 37–45. Disponible en: https://peerj.com/preprints/3190.pdf

[8] Pedregosa, F., et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.

[9] The Unicode Consortium. (2019). *The Unicode Standard, Version 12.0*. Disponible en: https://www.unicode.org/versions/Unicode12.0.0/

[10] Gray, J., & Reuter, A. (1992). *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann.

[11] Peña, D. (2002). *Análisis de datos multivariantes*. McGraw-Hill.