DOCTORADO EN CIENCIAS DE LA INGENIERÍA

DISEÑO DE UN MODELO DE GESTIÓN DE DATOS PARA ANALÍTICA EN LA INDUSTRIA

PROYECTO DE INVESTIGACIÓN III

Que presenta
Eduardo Antonio Hinojosa Palafox

Director
Dr. Oscar Mario Rodríguez Elías, ITH

Comité Revisor
Revisores internos
Dr. Héctor Guerra Crespo, IITG
Dr. Madain Pérez Patricio, ITTG
Dr. José Antonio Hoyo Montaño, ITH
Revisores externos
Dr. Jesús Horacio Pacheco Ramírez, UNISON
Dr. José Manuel Nieto Jalil, ITESM CSN

Diciembre 2019

28/11/2019

1. Introducción

1.1 El alcance de la gestión de datos en la industria

Los datos son un activo invaluable [1] que permite la manufactura inteligente. Su importancia estratégica es adquirir valor con significado específico a través del procesamiento especializado [2] (véase Fig. 1):

- Antes de que la manufactura comience, la planificación inteligente de la producción se lleva a cabo teniendo en cuenta los datos de los recursos de fabricación. Sobre la base de la relación de los datos globales, el programa de planificación global y optimizado podría generarse rápidamente, mejorando la velocidad y precisión de la planificación.
- En el proceso de fabricación, los datos en tiempo real deberían permitir la supervisión del proceso de fabricación, de modo que los fabricantes puedan mantenerse actualizados sobre los cambios para desarrollar estrategias óptimas de control operativo [3].
- Mantenimiento, Reparación y Operaciones (MRO) preventivo activo, mediante la recopilación y el análisis de datos masivos de dispositivos o productos inteligentes para el diagnóstico de fallas [4] y la optimización del proceso de operación.

Figura 1. Aplicación de datos de fabricación

Los datos de fabricación generalmente provienen de los siguientes aspectos [5]:

- Datos de recursos de fabricación, que incluyen a) datos de equipos recopilados de fábricas inteligentes por el Internet Industrial de las Cosas (IIoT, por sus siglas en inglés); b) datos de materiales y productos recopilados por sí mismos y por los sistemas de servicios; c) datos ambientales.
- Datos de gestión de sistemas de información de fabricación (MES, ERP, CRM, SCM y PDM) y sistemas asistidos por computadora (CAD, CAE y CAM).

Sin embargo, los datos en bruto son poco útiles (ver Tabla VI). Debe procesarse a través de varios pasos para extraer valor de ellos [37]. Primero, los datos se recopilan mediante diversas fuentes, como IIoT, PLC´s y SCADA, etc. Debido a las características de múltiples fuentes, heterogeneidad, multi escala, ruido y otros, los datos deben limpiarse antes de procesarse.

TABLA I. PROPIEDAD DE LOS DATOS Y LIMITACIONES

Propiedades de datos	Limitaciones
Muestreo	Los datos proporcionados se muestrean equidistantemente.
Volumen	Puntos de datos de gran volumen – (365 * 24 * 60 * sensores).
Veracidad	Los datos contienen valores atípicos, fallos del sensor y valores perdido.
Variedad	Los datos contienen valores atípicos, fallos del sensor y valores que faltan.
Redundancia	Los datos contienen sensores redundantes y altamente correlacionados.

1.2 Escenarios para analítica en Sistemas Ciber Físicos Industriales (SCFI)

Los análisis prescriptivos son inherentemente complejos en comparación con el análisis descriptivo y predictivo debido a la necesidad de alinear tecnología, modelado, pronósticos, optimización y la experiencia en el campo [6]. Por esto el Big Data en la industria está en sus primeras etapas [7] [8]. Para comprender los alcances de la analítica de Big Data en los SCFI proponemos diferentes escenarios que se describen a continuación (Ver Tabla II):

1) Integración de fuentes de datos en SCFI. La amplia variedad de sistemas ciber físicos industriales (SCFI) implementados en una fábrica inteligente genera enormes cantidades de datos. Sin embargo, dado que estos datos provienen de fuentes heterogéneas (PLC, SCADA, ERP), se requiere un sistema de extracción, transformación y carga (ETC) para la combinación, integración y posterior almacenamiento a gran escala.

2) Procesamiento de datos escalable y elástico. Para garantizar el procesamiento de Big Data con una latencia muy baja en tiempo real procedente de tecnologías IIoT (sensores inteligentes y RFID), la arquitectura de nube híbrida tiene que ser escalable y elástica.

3) La composición de los eventos basados en datos. A través del análisis prescriptivo se proporciona una estimación, de la influencia esperada de los parámetros de fabricación, a partir de datos en tiempo real (IIoT) en un tiempo de respuesta confiable.

4) Servicios de optimización. Proporcione un modelo de análisis predictivo a partir de tecnologías IIoT con procesamiento en nubes de Big Data hibridas en un tiempo de respuesta fiable.

5) Analítica embebida. Proporcione algoritmos específicos de análisis de datos adaptados al hardware embebido que produzca descubrimientos en una visión cercana al proceso/máquina específica, basada en datos generados propios y fuentes de datos estáticas, en un tiempo de respuesta fiable.

En la tabla II presentamos diferentes propuestas encontradas en la literatura agrupadas por cada escenario.

TABLE II. RETOS PARA LA ANALÍTICA EN SCFI

Escenario	Propuestas
Integración de fuentes de datos SCFI	[9] [10] [11]
Procesamiento de datos escalable y elástico	[12] [13] [14] [15] [16]
Composición de los eventos basados en datos	[17] [18] [19]
Servicios de optimización	[20] [21] [22] [23]
Analítica embebida	[24]

2. Aanalítica en tiempo real en Sistemas Ciber Físicos Industriales

2.1 Detección de anomalías basada en aprendizaje automático

La detección de anomalías en el análisis en tiempo real para SCFI ha permitido una nueva forma de optimizar los sistemas industriales ayudando a los analistas y operadores a resolver posibles problemas [9]: condiciones de proceso inusuales, características atípicas del producto, imágenes de defectos. A pesar de la enorme cantidad de datos disponibles, los eventos particulares de interés siguen siendo muy raros [10]. Estos eventos raros, a menudo llamados anomalías, se definen como eventos que ocurren con muy poca frecuencia (su frecuencia varía del 5% a menos del 0,01% dependiendo de la aplicación) [11]. El problema del análisis de anomalías ha sido ampliamente estudiado por los científicos de datos, el aprendizaje automático y la estadística [12]. Los métodos de aprendizaje supervisados generalmente crean un modelo de predicción para eventos anómalos basados en datos etiquetados (el conjunto de entrenamiento) y lo usan para clasificar cada caso [13]. Las principales debilidades de las técnicas de extracción de datos supervisadas incluyen la necesidad de tener datos etiquetados, que pueden llevar mucho tiempo para aplicaciones de la vida real, y la incapacidad para detectar nuevos tipos de eventos anómalos. Por otro lado, los métodos de aprendizaje no supervisados no necesitan datos etiquetados y detectan eventos como datos muy distintos de la mayoría de los datos basados en alguna medida [14]. Las técnicas de detección de anomalías no supervisadas se basan en suposiciones sobre valores atípicos frente al resto de los datos. Según las premisas que adoptemos, podemos clasificar los métodos de detección de anomalías en cuatro diferentes [15], en la figura 2 se muestra una clasificación de estos métodos.

Figura 2. Clasificación de los métodos no supervisados para la detección de anomalías

Los algoritmos de detección anomalías no supervisados pueden identificar nuevos tipos de eventos raros como desviaciones del comportamiento normal, pero sufren de una posible alta tasa de falsos positivos, principalmente porque los datos procesados también se reconocen como valores atípicos, y por lo tanto son marcados como un evento [16]. Un enfoque para resolver este problema es el uso de análisis de conjuntos para la detección de anomalías en espacios multidimensionales [17], el análisis de conjuntos se considera particularmente importante en escenarios de flujo de datos en los que los resultados de clasificadores individuales no son suficientes [18], los métodos de conjuntos exploran subespacios de los datos para descubrir anomalías [19]. Numerosos algoritmos han sido propuestos para la detección de valores atípicos no supervisados en los últimos años [14]. En [20] varios paquetes de detección de valores atípicos se mencionan: ELKI Data Mining [21], RapidMing [22] en Java y Valores Atípicos en R [23], los algoritmos de detección más populares se han implementado en PyOD, como se muestra en la Tabla III (nota: se han conservado sus nombres en ingles así como sus abreviaciones por ser conocidos de esta forma en la comunidad científica).

TABLA III. ALGORITMOS MÁS COMUNES PARA LA DETECCIÓN DE ANOMALÍAS. (pyod -https://github.com/yzhao062/pyod)

Categoría	Método	Descripción	Referencia
Modelo estadístico	**PCA**	Principal Component Analysis	[24]
Modelo estadístico	**MCD**	Minimum Covariance Determinant	[25] [26]
Modelo estadístico	**OCSVM**	One-Class Support Vector Machines	[27]
Proximidad	**LOF**	Local Outlier Factor	[28]
Proximidad	**COF**	Connectivity-Based Outlier Factor	[29]
Proximidad	**CBLOF**	Clustering-Based Local Outlier Factor	[30]
Proximidad	**LOCI**	Fast outlier detection using the local correlation integra	[31]
Proximidad	**HBOS**	Histogram-based Outlier Score	[32]
Proximidad	**kNN**	kNearest Neighbors	[33]
Proximidad	**AvgKNN**	Average kNN	[34]
Proximidad	**MedKNN**	Median kNN	[34]
Proximidad	**SOD**	Subspace Outlier Detection	[35]
Probabilística	**ABOD**	Angle-Based Outlier Detection	[36]
Probabilística	**FastABOD**	Fast Angle-Based Outlier Detection using approximation	[36]
Probabilística	**SOS**	Stochastic Outlier Selection	[37]
Redes neuronales	**AutoEncoder**	Fully connected AutoEncoder	[38]
Redes neuronales	**SO_GAAL**	Single-Objective Generative Adversarial Active Learning	[39]
Redes neuronales	**MO_GAAL**	Multiple-Objective Generative Adversarial Active Learning	[39]

2.2 Detección de anomalías en datos temporales

Los datos temporales contienen un conjunto de valores que normalmente se generan mediante la medición continua a lo largo del tiempo, por ejemplo, una empresa de manufactura recopila datos de sensores a través de diferentes aspectos, incluyendo la línea de productos, equipos de fabricación, proceso de fabricación y las condiciones ambientales. La continuidad temporal se refiere al hecho de que no se espera que los patrones en los datos cambien abruptamente a menos que haya procesos anormales en el funcionamiento [40]. En [41] se presenta una visión general completa y estructurada de las técnicas de detección de anomalías para datos temporales en series de tiempo discretas y en transmisión (ver Fig. 3). En las series de datos discretas, la asunción de la continuidad temporal desempeña un papel fundamental en la identificación de eventos raros [42], ya sean anomalías contextuales o colectivas [15]. Los valores atípicos son contextuales cuando los valores en marcas de tiempo específicas cambian repentinamente con respecto a sus valores adyacentes temporalmente, mientras que el evento raro es colectivo cuando series temporales completas o grandes subsecuencias dentro de una serie temporal tienen formas inusuales. En comparación con los datos estáticos, los datos de transmisión no tienen una longitud fija. Las dependencias temporales para secuencias multidimensionales se utilizan de manera diferente que en las series temporales, estos métodos están más cerca de los modelos multidimensionales convencionales, pero con un componente temporal, que representa la deriva temporal y las desviaciones. En comparación con los datos estáticos, los datos de transmisión no tienen una longitud fija. Las dependencias temporales para secuencias multidimensionales se utilizan de manera diferente que en las series temporales, estos métodos están más cerca de los modelos multidimensionales convencionales, pero con un componente temporal, que representa la deriva temporal y las desviaciones. Los flujos de sensores son una de las aplicaciones más comunes de detección de anomalías en datos temporales[43], este problema tiene doble aplicabilidad, tanto en términos de eliminación del ruido subyacente, como en términos de detección de eventos inusuales de la secuencia del sensor.

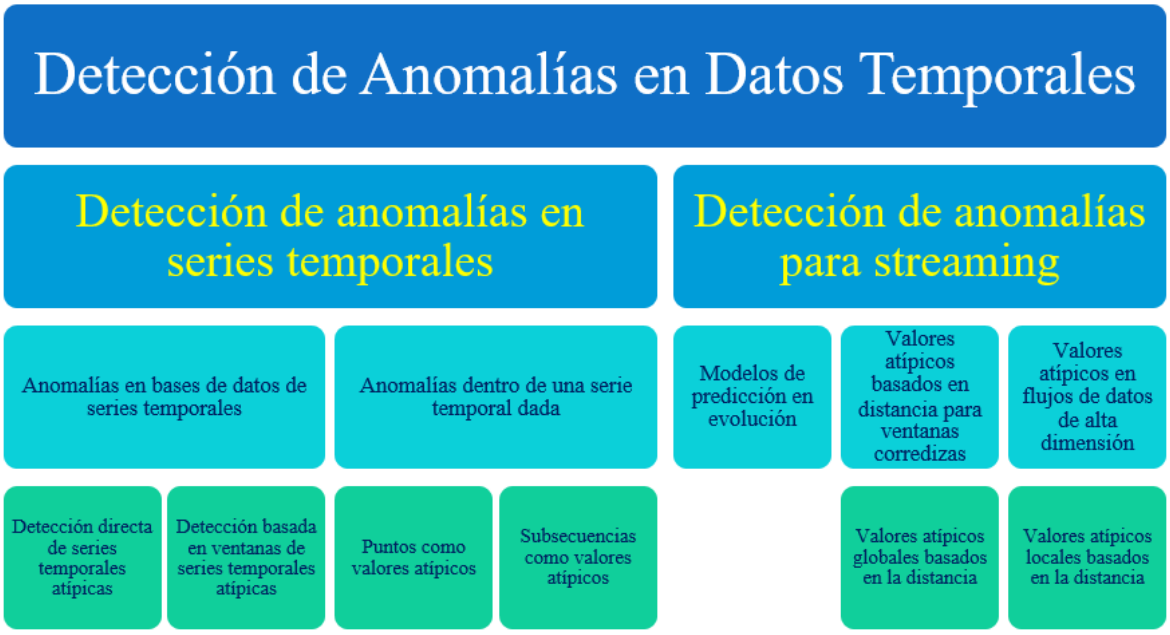

Figura 3. Clasificación de algoritmos para detección de anomalías en datos temporales

2.3 Aplicaciones para la detección de anomalías en la industria 4.0

En base al marco de referencia anterior, en la Tabla IV, se proponen diferentes escenarios industriales para la aplicación de analítica en tiempo real de acuerdo al origen de los datos y la serie temporal en el contexto de los SCIF en la industria 4.0.

TABLA III. ESCENARIOS PARA LA DETECCIÓN DE ANOMALÍAS

Escenario industrial	Escenario para analítica	Origen de los datos	Dato temporal	Conjunto de datos
Equipos de instalaciones	Servicios de datos de optimización	Datos de sensores y entornos	Datos de transmisión	**Conjunto de datos de la central de energía de ciclo combinado**
Equipos de proceso	Analítica embebida para Servicios de datos de optimización	Datos de detección de fallos del sensor	Datos de transmisión	**Sistema de producción versátil** Datos para monitoreo de condiciones, mantenimiento predictivo, aprendizaje automático. **PHM Data Challenge 2015** Operación de una planta y la capacidad de detectar eventos de fallo por adelantado.
Resultados del proceso	La composición de los eventos basados en datos	Historial de procesos y mediciones	Serie temporal	**UCI SECOM** Dataset (Conjunto de datos del proceso de fabricación de semiconductores) - Selección y categorización de características de proceso de fabricación **Rendimiento de la línea de producción de Bosch** - Reducir las fallas de fabricación **Fabricación más ecológica de Mercedes-Benz** ¿Puede reducirse el tiempo que un Mercedes-Benz pasa en el banco de pruebas?
Defectos físicos	Analítica embebida	Imágenes y características de defectos	Datos de transmisión	**Desgaste de la herramienta de molino CNC** Datos de mecanizado CNC de variación **Impresora 3D** Conjunto de datos para ingenieros mecánicos.
Producto	Servicios de datos de optimización	Características de la prueba del producto	Serie temporal	**Predicción de calidad en un proceso minero**

En la sección 4 se desarrolla el escenario de equipos de proceso, tomando como caso de uso la operación de una planta y la capacidad de detectar eventos de fallo, utilizando para ello el conjunto de datos del PHM Data Challenge 2015, organizado por la "Prognostics and Health Management Society" (phmsociety - https://www.phmsociety.org/), una organización sin fines de lucro dedicada al avance de PHM como disciplina de ingeniería.

3. Técnicas de evaluación para algoritmos no supervisados

3.1 Fundamentos teóricos

El proceso de modelado en algunos problemas como la detección de anomalía suele ser un proceso inherentemente subjetivo, donde la función objetiva o el modelo definido para un problema determinado depende de la comprensión del comportamiento de los datos [44]. Por ejemplo, el algoritmo del vecino más cercano para anomalías podría proporcionar resultados muy diferentes al algoritmo de una máquina vectorial de soporte de una clase, debido a las diferencias subyacentes en las asunciones que estos modelos tienen. Por otro lado, el modelo seleccionado puede ser extremadamente sensible a la elección de parámetros utilizados en la detección de anomalías. Todos estos problemas a menudo hacen que la evaluación de la calidad de los algoritmos de detección de valores atípicos sea más difícil, y en ausencia de la etiqueta de verdad, existe incertidumbre sobre la verdadera eficacia en la selección de un algoritmo [15].

La mayoría de los algoritmos de detección de anomalías generan puntuaciones para cuantificar la "valoración atípica" de los puntos de datos. Una vez calculadas las puntuaciones, se pueden convertir en etiquetas binarias.

Considere una instancia de datos denotada por $\bar{X}_i$, para la que la puntuación de valores atípicos se modela utilizando los datos de entrenamiento $\mathcal{D}$. Asumimos que todos los puntos de entrenamiento de datos son

generados por una misma distribución base. La puntuación ideal se obtiene por una función desconocida f($\bar{X}_i$) y se asume que las puntuaciones generadas por esta función ideal también satisfacen la media cero y la suposición de varianza unitaria sobre todos los puntos posibles generados por la distribución de datos base:

$$y_i = f(\bar{X}_i) \quad (1)$$

Dado que se desconoce el modelo verdadero f(·), la puntuación atípica de un punto de prueba $\bar{X}_i$ sólo se puede estimar con el uso de un modelo de detección de valores atípicos g ($\bar{X}_i, \mathcal{D}$) utilizando el conjunto de datos base $\mathcal{D}$. Por ejemplo, en los detectores de valores atípicos de los vecinos más cercanos, la función $g(\bar{X}_i, \mathcal{D})$ se define de la siguiente manera:

$$g(\bar{X}_i, \mathcal{D}) = \alpha KNN - distance(\bar{X}_i, \mathcal{D}) + \beta \quad (2)$$

Tenemos que, α and β son constantes que son necesarias para estandarizar las puntuaciones la media a cero y la varianza a la unidad con el fin de respetar la restricción en la interpretación absoluta de las puntuaciones atípicos. Por otro lado, la función g ($\bar{X}_i, \mathcal{D}$) no modela correctamente la verdadera función f ($\bar{X}_i$), por lo que se generan errores. Esto se conoce como sesgo de modelo. Una segunda fuente de error es la varianza. La varianza es causada por el hecho de que la puntuación de valores atípicos depende directamente de la creación de instancias específica del conjunto de datos $\mathcal{D}$. Ahora, Sea $\mathcal{D}$ los datos de entrenamiento, y $\bar{X}_1$ ... $\bar{X}_n$ ser un conjunto de *n* puntos de prueba cuyas puntuaciones de valores atípicos (hipotéticamente ideales pero no observados) son y_1 ... y_n. Utilizamos un algoritmo de detección de valores atípicos no supervisado que utiliza la función $g(\cdot)$ para estimar estas puntuaciones. Por lo tanto, las puntuaciones resultantes de $\bar{X}_1$ ... $\bar{X}_n$ utilizando los datos de entrenamiento $\mathcal{D}$ son $g(\bar{X}_1, \mathcal{D})$… $g(\bar{X}_n, \mathcal{D})$, respectivamente. El error medio cuadrado, o EMC, de los detectores de los puntos de prueba sobre una realización particular $\mathcal{D}$ de los datos de entrenamiento se obtiene mediante el promedio de los errores al cuadrado en diferentes puntos de prueba [44]:

$$E[EMC] = \frac{1}{n}\sum_{i=1}^{n}\{y_i - g(\bar{X}_i, \mathcal{D})\}^2 \quad (3)$$

El EMC esperado, sobre diferentes realizaciones de los datos de entrenamiento, generados usando algún proceso aleatorio, es el siguiente:

$$E[EMC] = \frac{1}{n}\sum_{i=1}^{n} E[\{y_i - g(\bar{X}_i, \mathcal{D})\}^2] \quad (4)$$

El término en el corchete en el lado derecho de la Ecuación 4 se puede volver a escribir de la siguiente manera:

$$E[EMC] = \frac{1}{n}\sum_{i=1}^{n} E[\{y_i - f(\bar{X}_i) + f(\bar{X}_i) - g(\bar{X}_i, \mathcal{D})\}^2] \quad (5)$$

Se puede mostrar lo siguiente:

$$E[EMC] = \frac{1}{n}\sum_{i=1}^{n} E[\{f(\bar{X}_i) - g(\bar{X}_i, \mathcal{D})\}^2] \quad (6)$$

Este lado derecho se puede descomponer aún más añadiendo y restando $E[g(\bar{X}_i, \mathcal{D})]$ dentro del término cuadrado:

$$\begin{aligned} E[EMC] = {} & \frac{1}{n}\sum_{i=1}^{n} E[\{f(\bar{X}_i) - E[g(\bar{X}_i, \mathcal{D})]\}^2] \\ & + \frac{1}{n}\sum_{i=1}^{n}\{f(\bar{X}_i) - E[g(\bar{X}_i, \mathcal{D})]\}\{E[g(\bar{X}_i, \mathcal{D})] - E[g(\bar{X}_i, \mathcal{D})]\} \\ & + \frac{1}{n}\sum_{i=1}^{n} E[\{E[g(\bar{X}_i, \mathcal{D})] - g(\bar{X}_i, \mathcal{D})\}^2] \quad (7) \end{aligned}$$

Tenemos:

$$E[EMC] = \frac{1}{n}\sum_{i=1}^{n} E[\{f(\bar{X}_i) - E[g(\bar{X}_i, \mathcal{D})]\}^2] + \frac{1}{n}\sum_{i=1}^{n} E[\{E[g(\bar{X}_i, \mathcal{D})] - g(\bar{X}_i, \mathcal{D})\}^2]$$

$$= \frac{1}{n}\sum_{i=1}^{n}\{f\,(\bar{X}_i)\; - E[g\,(\bar{X}_i, \mathcal{D})]\}^2 + \frac{1}{n}\sum_{i=1}^{n} E[\{E[g\,(\bar{X}_i, \mathcal{D})]\; -\; g\,(\bar{X}_i, \mathcal{D})\}^2] \quad (8)$$

El primer término de la expresión antes mencionada es el sesgo (cuadrado), mientras que el segundo término es la varianza. Dicho simplemente, se obtiene lo siguiente:

E[EMC] – Sesgo^2 + Varianza

3.2 Algoritmo para la selección del modelo

Con base en base a lo anterior, a continuación nuestra propuesta para selección del mejor modelo basado en la reducción del sesgo y varianza de cada método evaluado.

Input: Lista de detectores base *B*, Lista de parámetros *P*, dataset *D*
Output: TS, RB (para cada algoritmo)

Inicializa la lista de detectores base *B*
Permite a TS, *P, y RB,* ser una columna del dataset *D*
Inicializa *TMS* como un modelo de series temporales en *D*.
Construye la serie de tiempo *TS* usando *TMS*
for cada *B* **do**
  Inicializa los parámetros *P* del algoritmo *B*
  for cada *P* ***do***
    Identifica el rango *min, max*
    Construye una lista i con valores aleatorios de [*min, max*]
    for each *i* **do**
      Aplica el algoritmo *B* a TS con el parámetro *i* para construir el modelo *M*
        Obtén la predicción de *M(i)*
        Obtén los puntajes estandarizados *S(i)*
      Obtén MES (*S*(i))
    end for
      Selecciona el **Mínimo** MES (*puntajes estandarizados P)* para *P*
  end for
  Selecciona el **Mínimo** MES (*puntajes estandarizados B)*
end for
Selecciona el Mínimo MES (*B*)
Muestra *el mejor modelo de B*

Sea $D \in R^{n\times d}$ denote los datos con *n* puntos y *d* características, el algoritmo primero genera un grupo de detectores de base $B = \{B_1, ..., B_r\}$ Inicializa con un rango de hiperparámetros. Todos los detectores de base se entrenan y a continuación, la inferencia se realiza en el mismo conjunto de datos *D*. Los resultados se integran en una matriz de puntajes de anomalías $O(D)$, $[B_1(D), ..., Br(D)] \in R_{n\times R}$ donde $B_r(\cdot)$ denota el vector de puntuación del r^{th} detector base. Cada detector de puntuación $B_r(D)$ se normaliza mediante la *normalización Z*. MES mide la competencia de cada detector de base por el error medio cuadrado, El detector B_r con la menor varianza y sesgo se considera el detector más competente.

4. Detección de eventos de fallo en planta industrial

El diagnóstico de fallas tiene un papel crítico en los sistemas de plantas industriales. Un sistema de diagnóstico de fallas robusto y preciso ayuda a prevenir accidentes fatales, ahorra costos y aumenta la eficiencia de la producción [45]. El desarrollo de un sistema de diagnóstico de fallas de alto rendimiento para un sistema en particular requiere principalmente dos tipos de información: (1) una comprensión profunda del sistema objetivo o (2) datos de monitoreo de condición / registro de fallas. Un amplio nivel de conocimiento sobre fallas del sistema (es decir, mecanismos, causas fundamentales) puede facilitar el diagnóstico efectivo de fallas para los sistemas de plantas industriales. Por otro lado, una cantidad significativa de monitoreo de las fallas a través de los datos de registro, si están disponibles, pueden proporcionar información excelente para el diagnóstico basado en datos (por ejemplo, analítica de Big Data). Desafortunadamente, tener un conocimiento profundo del sistema a optimizar es casi imposible en sistemas reales en plantas industriales, debido a que tales sistemas están compuestos de numerosos componentes y operan en una variedad de condiciones. Sin embargo, la mayoría de los datos disponibles contienen registros de faltas incompletos o faltantes debido a factores humanos o sistemas de monitoreo que proporcionan datos deficientes (por ejemplo, formato obsoleto).

El propósito de esta sección es desarrollar una aplicación de detección de anomalías para eventos de fallo utilizado el algoritmo para la selección del modelo no supervisado presentado en la sección 3.2, además se establece una comparativa con los métodos supervisados de clasificación descritos en la sección 4.2.

4.1 Planteamiento del problema

La sociedad de pronósticos y gestión de la salud (PHM) abordó el tema del diagnóstico de fallas de planta industrial con sistemas con datos de registro de fallas incompletos en la Competencia de Desafío de Datos PHM 2015 [45]. El problema en esta competencia fue identificar (1) los tipos de fallas y (2) los tiempos de inicio y finalización de las fallas correspondientes. El problema refleja situaciones del mundo real porque los registros de fallas a menudo faltan en aplicaciones industriales reales del mundo real. Como se aprecia en la figura 4, los datos dados representan: a) series temporales de mediciones de sensores y señales de referencia de control para cada uno de varios componentes de control de la planta (por ejemplo, 6 componentes); (b) datos de series temporales que representen mediciones adicionales de un número fijo de zonas de la planta durante el mismo período de tiempo (por ejemplo, 3 zonas), donde una zona puede abarcar uno o más componentes de la planta; (c) eventos de fallas de planta, cada uno caracterizado por una hora de inicio, una hora de finalización y un código de error. Cada planta es específica a través de su número de componentes y el número de zonas. Sin embargo, cada planta registra errores del mismo conjunto fijo de errores. Solo los errores de tipo 1-5 son de interés, mientras que el código 6 representa todos los demás errores que no están enfocados. La frecuencia de las mediciones es de aproximadamente de una muestra cada 15 minutos, y los datos de la serie temporal abarcan un período de aproximadamente tres a cuatro años.

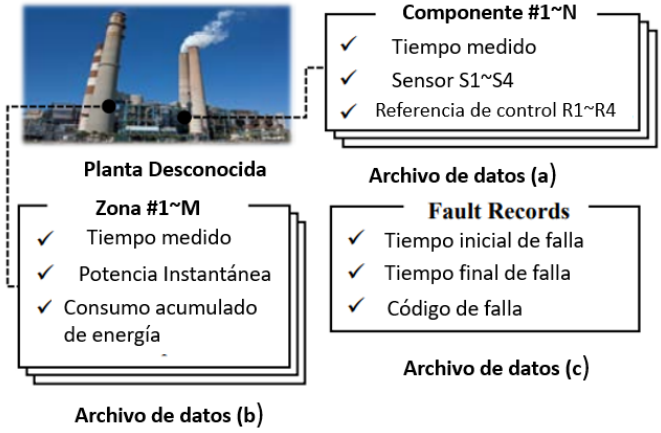

Figura 4. Descripción de los conjuntos de datos proporcionados (adaptado de [46])

La tarea consiste en predecir eventos de error futuros de los tipos 1-5 y el momento de su aparición desde datos anteriores. Por ejemplo, el conjunto de datos para la Planta #1 se proporciona mediante una colección de

tres archivos [.csv]: plant-1a.csv, plant-1b.csv, plant-1c.csv. Cada uno de los archivos (a), (b) y (c) contiene información como se ha descrito anteriormente. Más precisamente las columnas de cada uno de los archivos (a), (b) y (c) [.csv] son:

a) Mediciones de la planta por componente: Número de componente "m", tiempo "t", sensores "S1"- "S4", y referencias de control "R1"-"R4".
b) Mediciones adicionales de la planta por zona en la planta: número de zona "n", hora "t", sensores "E1" y "E2".
c) Errores: Hora de inicio "t1", hora de finalización "t2" y código de error "F".

- Cada componente de la planta está controlado por un sistema de bucle de retroalimentación como se representa en la figura siguiente; los componentes de la planta están desarticulados.
- Cada zona mide la energía acumulada (E1) y la energía instantánea (E2) en secciones desarticuladas de la planta que cubren uno o más componentes.
- Los errores son independientes entre sí. Además, un error F es independiente de los datos fuera de un período de tiempo de tres horas antes de la hora de inicio del error.

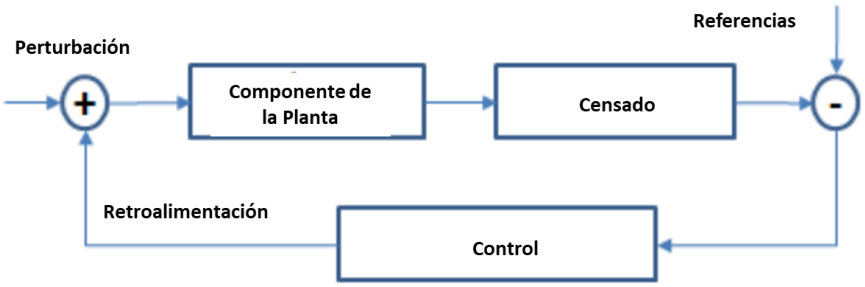

Figura 3. Sistema de bucle de retroalimentación

4.2 Clasificación de eventos de falla

En el desafío presentado en la sección anterior, se cuenta con un archivo completo de datos de los eventos de error que puede ser usado para probar varios modelos con los datos de entrenamiento de validación cruzada y, a continuación, evaluar su rendimiento en función de su capacidad para pronosticar errores en los datos de prueba de validación cruzada. En la tabla VI se presenta un resumen del enfoque y metodología propuestos por los ganadores del primero, segundo y tercer lugar, respectivamente los autores [47], [46] y [48].

TABLA VI. ARTÍCULOS DE LA CONFERENCIA PHM 2015

Autores	Algoritmo	Metodología
[47]	Se trataron varios algoritmos de aprendizaje automático: vecinos más cercanos al K (KNN), bahías ingenuas, máquina de refuerzo de gradiente (GBM), bosque aleatorio, regresión logística penalizada, etc. En el algoritmo final, los autores usaron máquina de refuerzo de gradiente, bosque aleatorio y regresión logística penalizada.	Probar varios modelos utilizando los datos de entrenamiento de validación cruzada y, a continuación, evaluar su rendimiento en función de su capacidad para pronosticar errores en los datos de prueba de validación cruzada.
[46]	Los autores proponen una técnica de recuperación de registro de errores para el diagnóstico de errores.	Extraer las características relevantes en función de la interpretación física de los datos. A continuación, proponer un clasificador basado en el análisis discriminante de Fisher (FDA, en inglés) para incorporar datos incompletos.
[48]	Los autores adoptaron el clasificador de árbol de decisión para predecir el tipo de falla. Más específicamente, fue usado el Bosque aleatorio y el Árbol de decisión de aumento de gradiente como clasificadores.	El primer paso es extraer características útiles de los datos sin procesar para facilitar la detección. El segundo paso es construir el modelo del clasificador de árbol de decisión. Tercer paso es desarrollar el algoritmo.

En la siguiente sección abordamos la solución al problema presentado desde la perspectiva de la detección de anomalías no supervisadas, por lo que se asume no existe un archivo de eventos de error, lo que implica el no poder clasificar el tipo de falla que sucederá, tampoco mejorar el rendimiento del modelo, ni seleccionar el

mejor a través del entrenamiento y validación cruzada. Para ello usamos el algoritmo para la selección propuesto en la sección 3.2.

4.3 Descripción de datos y procesamiento

Comenzamos nuestro análisis estudiando primero los datos para obtener cualquier información que sea de utilidad para el modelo de detección de anomalías, para esto es importante comprender qué tipo de variables existen, el primer paso es extraer características útiles de los datos sin procesar para facilitar la detección, por esto, en la Tabla IV se muestra el conteo de los niveles únicos de las variables categóricas de las primeras cinco plantas, como se observa la mayoría de las plantas tienen un número irregular de componentes y zonas. Como resultado, puede haber cientos de reglas que definen fallas de estos sistemas basadas en la combinación de múltiples señales de varios componentes y zonas. En este caso, es imposible identificar reglas generales para el diagnóstico de la mayoría de los errores.

TABLA IV. CUENTA DE NIVELES ÚNICOS PARA VARIABLES CATEGÓRICAS

Planta	Nm	Nn	S3	R1	R2	R3	R4
1	6	3	12	38	6	8	3
2	13	2	11	26	6	6	3
3	10	2	12	30	7	8	3
4	8	4	12	34	7	7	3
5	3	2	12	12	7	6	3

Nm: Número de Componentes; Nn: Número de Zonas; S3 (Sensor 3), (R1~R4 Referencias de Control

En la figura IV se muestra las características las series temporales en sensores 1, 2 y 4, y en la energía eléctrica 1 y 2 durante un poco más de dos años para la planta 1.

Figura 4. Serie temporal para las variables S1~S3, E1 y E2 en la planta 1

Para incluir las variables categóricas en el procesamiento de datos, a partir de estas se crearon variables tontas, de tal forma que de la variable categórica Nn se obtuvieron 6 variables tontas (demand 1~5 y Meter), de la variable Nm se obtuvieron 14 variables tontas (HVAC 1~14) y las variables tontas Off, Occupied y set back se derivaron de la variable categórica R4.

TABLA V. VARIABLES TONTAS

Nn	Nm	R4
Demand 1~5	HVAC 1~14	OFF
METER		Occupied Set back

En la figura 5 se muestra el resultado de haber procesado los archivos de las plantas 1~30 para corregir la fecha, los datos faltantes y crear las variables tontas.

```
                      time   S1   S2   S3   S4   R1  R2  R3   planta  \
0         2009-08-18 18:00:00  711  630   69  600  689  20  40   plant1
1         2009-08-18 18:00:00  725  460  101  705  689  20  40   plant1
2         2009-08-18 18:00:00  711  505   69  678  689  20  40   plant1
3         2009-08-18 18:00:00  705  630   69  600  689  20  40   plant1
4         2009-08-18 18:00:00  734  516  101  671  689  20  40   plant1
...                       ...  ...  ...  ...  ...  ...  ..  ..      ...
17192923  2012-11-09 22:00:00  734  511   68  674  700  40  40  plant30
17192924  2012-11-09 22:00:00  729  648   69  589  700  40  40  plant30
17192925  2012-11-09 22:00:00  736  734    0  534  700  40  40  plant30
17192926  2012-11-09 22:00:00  736  592   68  624  700  40  40  plant30
17192927  2012-11-09 22:00:00  736  714    5  547  700  40  40  plant30

          m_HVAC1  ...  m_HVAC3  m_HVAC4  m_HVAC5  m_HVAC6  m_HVAC7  m_HVAC8  \
0               1  ...        0        0        0        0        0        0
1               0  ...        0        0        1        0        0        0
2               0  ...        0        0        0        1        0        0
3               1  ...        0        0        0        0        0        0
4               0  ...        0        0        0        0        0        0
...           ...  ...      ...      ...      ...      ...      ...      ...
17192923        0  ...        1        0        0        0        0        0
17192924        0  ...        0        0        0        0        0        0
17192925        0  ...        0        1        0        0        0        0
17192926        0  ...        0        0        1        0        0        0
17192927        0  ...        0        0        0        1        0        0

          m_HVAC9  R4_OFF  R4_Occupied  R4_Setback
0               0       0            1           0
1               0       0            1           0
2               0       0            1           0
3               0       0            1           0
4               0       0            1           0
...           ...     ...          ...         ...
17192923        0       0            1           0
17192924        0       0            1           0
17192925        0       0            1           0
17192926        0       0            1           0
17192927        0       0            1           0

[17192928 rows x 26 columns]
```

Figura 5. Preprocesamiento de los datos de treinta plantas para los archivos tipo “a”

4.4 Condiciones de proceso inusuales

En esta sección se presenta el enfoque y la metodología utilizada para el monitoreo de procesos basado en aprendizaje máquina usando un modelo multivariado como base para la detección de anomalías en condiciones de proceso inusuales. El enfoque general presentado consta de dos partes: preprocesamiento y la aplicación del modelo al proceso en marcha para detectar valores atípicos en componentes subyacentes y dimensiones virtuales. La Figura 6 proporciona una visión general del proceso implementado. Con el modelo específico creado se calcula el valor esperado T, a continuación, una serie de errores E se calculan comparando el valor esperado con el valor real en el momento T y genera las anomalías más probables.

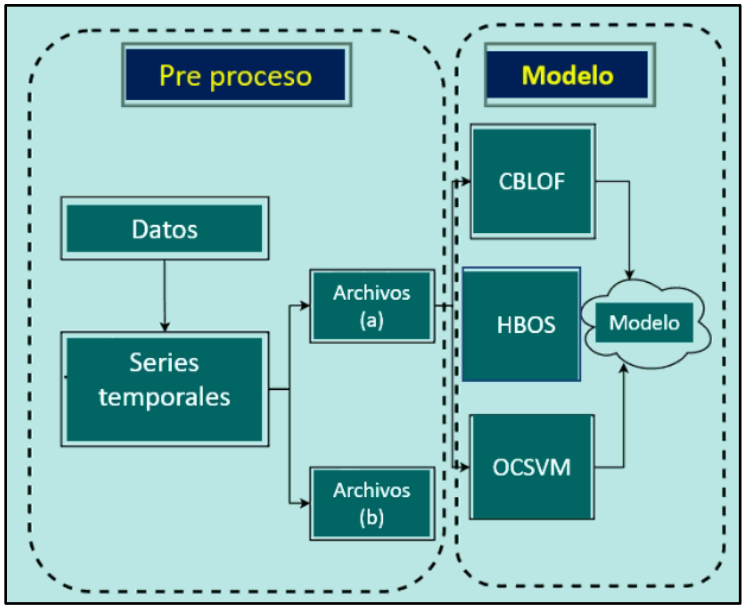

Figura 6. Gráfico de flujo general

Después del pre procesamiento, dividimos los datos en dos partes: datos de entrenamiento y datos de predicción. Entrenamos los algoritmos CBLOF, HBOS y OCSVM (ver la tabla III en la sección 2.1 para más información de estos métodos de detección) en su capacidad de detectar errores y el algoritmo de selección del modelo determina cuál es el modelo ganador, el cual es aplicado a los datos de predicción. Para cada planta se creó un modelo independiente para predecir un evento inusual. La tabla VI muestra los resultados obtenidos.

TABLA VI.

Planta	Archivos (A)				Archivos (B)			
	Tamaño	Anomalía	% Anomalía	Tiempo(s)	Tamaño	Anomalía	% Anomalía	Tiempo(s)
1	672530	65291	9.70	45207	339494	20161	5.93	4091
2	635660	45518	7.10	46183	363658	28924	7.95	4869
3	1049547	163469	15.57	68534	210313	12619	6.00	1368
4	824370	72817	8.83	39864	412247	31686	7.68	5899
5	283661	17980	6.33	8566	193343	11601	6.00	1179

Para poder visualizar las anomalías encontradas basadas en el algoritmo de selección de modelo, procedemos a ajustar las 24 dimensiones del archivo “a” mediante el método del análisis de componentes principales (PCA por sus siglas en inglés) para reducir el número a dos dimensiones y, en la figura 6, se visualizan los resultados, a través de una gráfica 2D que nos proporcione una imagen clara de los puntos de anomalías, las anomalías se resaltan como bordes rojos y los puntos normales se indican con puntos verdes en el trazado. Para poder apreciar más claramente la gráfica, solamente se presenta una muestra de los primeros 100 puntos del conjunto de datos del archivo “a”.

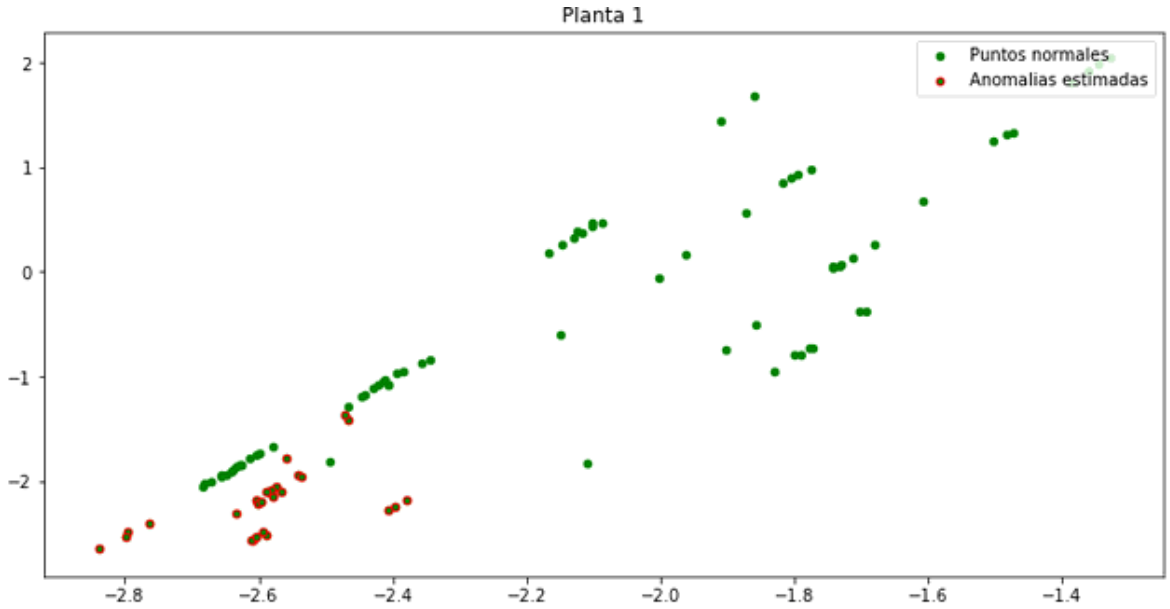

Figura 7. Gráfico de flujo general (100 puntos del archivo “a”)

5. Arquitectura de gestión de datos

Para satisfacer las necesidades que hemos presentado en la sección 1.2, capaz de manejar una amplia gama de cargas de trabajo y casos de uso, y en los que se requiere baja latencia de lectura y escritura, adoptamos la arquitectura Lambda como arquitectura de referencia [49] (ver Figura. 8). Este tipo de arquitectura de Big Data resuelve el problema de calcular las funciones en datos en tiempo real desglosando el problema en tres capas:

- Una capa por lotes que administra un conjunto de datos maestros inmutable y solo calcula las funciones de consulta denominadas lotes de vistas;
- Una capa de servicio que indexa las vistas por lotes para consultas ad hoc de baja latencia;
- Y una capa de velocidad que utiliza algoritmos rápidos e incrementales solo en datos recientes.

A continuación, se describe una propuesta para una guía de diseño de arquitecturas de gestión de datos para la industria 4.0, la figura 9 muestra la arquitectura propuesta y sus capas y componentes. Esta propuesta se beneficia del estado del arte, ya sea en la identificación de sus componentes principales y en la identificación de las tecnologías de Big Data que se adoptarán.

5.1 Capa por lotes

La capa por lotes es la primera en ser explicada, el componente de datos de recursos de fabricación representa a todos los productores de Big Data, un conjunto inmutable, de solo anexar de datos sin procesar, esto permite observar, los datos de entrada sin cambios y volver a procesarlos cuando hay cabio de criterios. El componente de procesamiento de datos gestiona datos con baja velocidad y simultaneidad (como, por

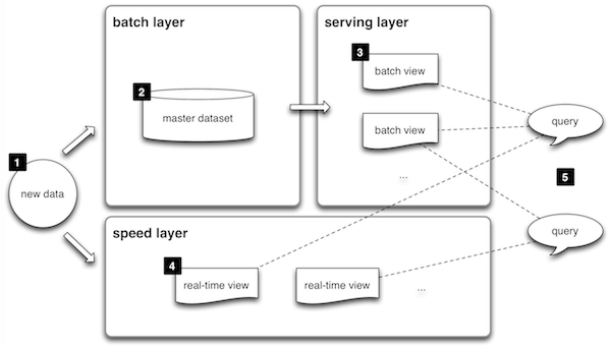

Fig. 8: Arquitectura Lambda(Fuente Lambda architecture - http://lambda-architecture.net/)

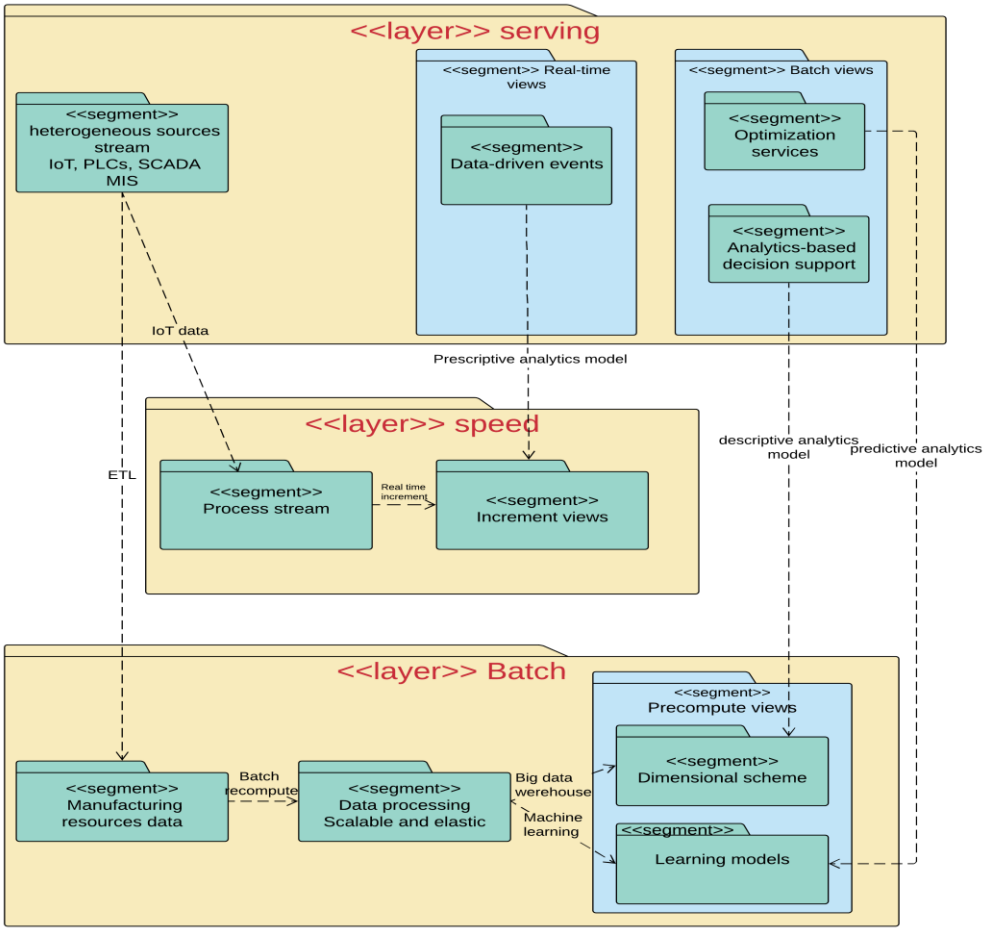
<<layer>> serving
<> heterogeneous sources stream IoT, PLCs, SCADA MIS
<> Real-time views
<> Data-driven events
<> Batch views
<> Optimization services
<> Analytics-based decision support
IoT data
Prescriptive analytics model
descriptive analytics model
predictive analytics model
ETL
<<layer>> speed
<> Process stream
Real time increment
<> Increment views
<<layer>> Batch
<> Precompute views
<> Dimensional scheme
<> Manufacturing resources data
Batch recompute
<> Data processing Scalable and elastic
Big data werehouse
Machine learning
<> Learning models

Fig. 9: Arquitectura de gestión de datos para la industria 4.0

ejemplo, datos de lecturas periódicas de bases de datos), o datos con un alto grado de velocidad y simultaneidad (como, por ejemplo, flujos de datos de dispositivos electrónicos y sensores). En el esquema dimensional, el componente es la pieza central en inteligencia empresarial y análisis, en un contexto como el representado por los grandes almacenes de datos, la tarea de modelado de datos se ve en otra perspectiva, por el modelo de datos multidimensionales (MDM) que permiten el análisis de datos desde diferentes perspectivas en el apoyo a los procesos de toma de decisiones. En el componente de aprendizaje de modelos, los parámetros que representan el modelo dependen de los algoritmos utilizados. En nuestro caso de uso, esto incluiría agregados temporales de fabricación de operaciones de datos para la detección de anomalías. En el contexto del procesamiento de grandes cantidades de datos, es importante paralelizar los pasos de procesamiento en función en gran medida de los datos y algoritmos reales utilizados.

5.2 Capa de velocidad

La capa de velocidad recibe el flujo de nuevos datos de IIoT. El componente de flujo de proceso es configurado para recibir datos de serie temporal con las características de la secuencia de datos sin procesar. El componente de vistas de incremento implementa la detección de anomalías de aprendizaje automático en el período de tiempo coincidente y produce un evento en la capa de servicio si se supera el umbral y se detecta consecutivamente durante un tiempo determinado.

5.3 Capa de servicio

La capa de servicio proporciona una vista de la salida de patrones de datos generados por los trabajos de capa por lotes y de velocidad. Las vistas en tiempo real analizan el flujo de datos entrante en tiempo real mediante el componente de eventos controlados por datos. Contiene los parámetros del modelo calculados en el componente de vistas de incremento; el modelo de anomalía que caracteriza las distribuciones de carga de las operaciones de fabricación agrega datos durante el período que se utiliza para el aprendizaje de modelos permitiendo el análisis de la información en tiempo real. Las vistas por lotes se utilizan para presentar el diagnóstico y el análisis predictivo; el componente de servicios de optimización mejora el diagnóstico del sistema de fabricación con una previsión de los Indicadores clave de rendimiento del negocio (KPIs, por sus siglas en inglés) basada en métodos de Big Data implementados en el componente de modelos de aprendizaje en la capa por lotes. El componente de apoyo a la toma de decisiones basado en análisis admite el análisis OLAP para informar y observar y muestra cuán grande o pequeño es el problema mediante el uso de un almacén de Big Data desde la capa de lotes.

5.3 Vista de implementación

Hoy en día, está surgiendo tecnologías diferentes para el procesamiento de datos, a menudo con funcionalidades superpuestas y requisitos de atributos de calidad. Para los arquitectos de sistemas, la selección de la tecnología de implementación es una tarea difícil que requiere la consideración de muchos detalles de implementación y restricciones de compatibilidad. La vista de implementación (ver Figura 10) proporciona una asignación coherente entre tecnologías y componentes funcionales especificados en el diseño de arquitectura.

Estas tecnologías son accesibles bajo una licencia de código abierto sin limitaciones de uso (ver Tabla VII).

TABLA VII. ASIGNACIÓN DE COMPONENTES A TECNOLOGÍAS.

Componente	Mapeo de tecnologías
Procesamiento de datos	Integración de datos— Spark streaming
Vistas por lotes	Visualización de datos— Grafana
Datos de recursos de fabricación	Sistema de archivos distribuido— Apache Hadoop (HDFS)
Esquema dimensional	Almacén de datos distribuido— Apache Hive
Modelos de aprendizaje	Marco de procesamiento de datos distribuido— Apache Spark (computación distribuida)
Fuentes heterogéneas de flujo de datos en tiempo real	Servicio de mensajería— Apache Kafka
Vistas en tiempo real	Herramientas de desarrollo— Módulos y bibliotecas de Python

Los datos de dispositivos y sistemas se integran mediante Apache Kafka y se transfieren mediante el protocolo MQTT a una plataforma de análisis como servicio en la nube (PaaS), donde son ingeridos por Spark Streaming, que admite el procesamiento por lotes y procesamiento de flujo de datos.

- *Procesamiento por lotes.* Spark Streaming interactúa con Kafka para obtener los datos disponibles a través de los productores de transmisión de datos. A continuación, los flujos de datos se pueden obtener, transformar y cargar (ETL, por sus siglas en inglés) en el componente de datos de recursos de fabricación. Los datos se almacenarán en una perspectiva histórica en Hadoop y se pueden almacenar en HDFS, un sistema de archivos distribuido para almacenar grandes volúmenes. Una vez que los datos estén disponibles, se pondrán a disposición para el análisis de datos a través de Hive en un almacén de datos en contexto de Big Data. Sin embargo, Hive se basa en HDFS que permite el almacenamiento y el procesamiento distribuidos, para almacenar y agregar grandes volúmenes de datos. Por otro lado, el componente del modelo de aprendizaje utiliza los datos almacenados en Hadoop para crear análisis predictivos mediante algoritmos de aprendizaje automático con Spark, una herramienta de procesamiento en tiempo real que permite un mejor rendimiento mediante la memoria RAM.
- *Procesamiento en tiempo real*. Una vez más, Spark Streaming interactúa con Kafka para obtener los datos disponibles a través de IIoT y ofrece procesamiento de flujo escalable, de alto rendimiento y tolerante a errores de secuencias de datos en vivo para la detección de valores atípicos. A continuación, los datos se

almacenan o insertan en el componente de eventos controlado por datos para su análisis en tiempo real. En

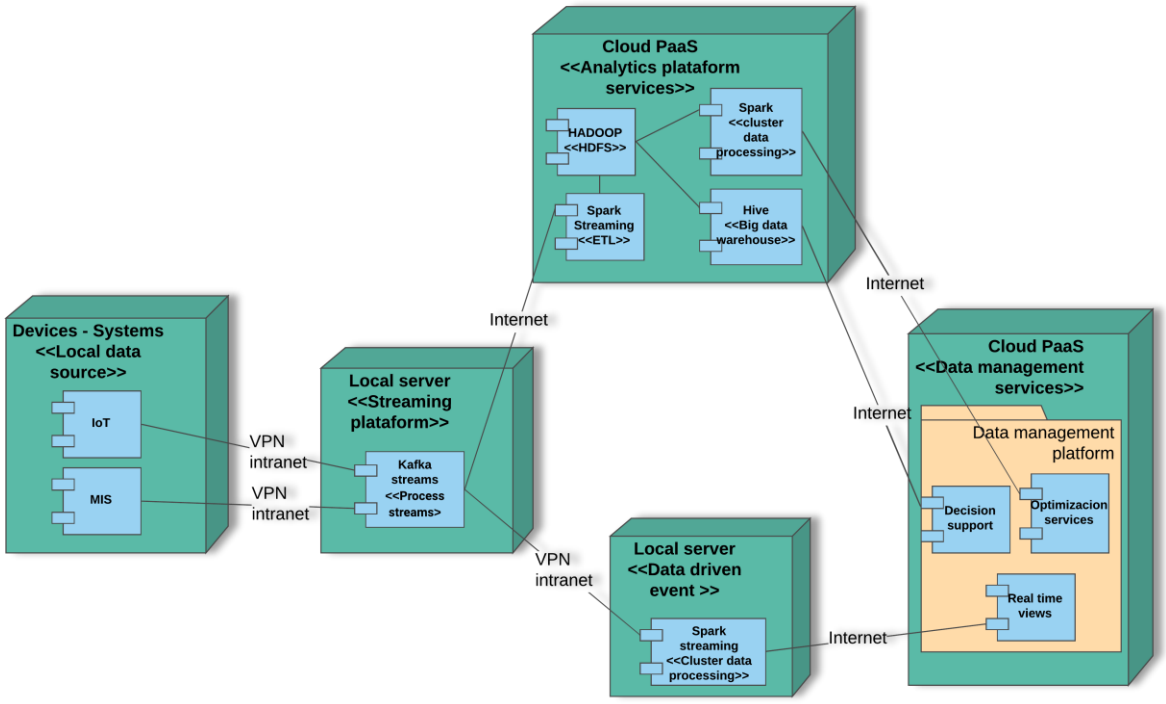

Fig. 10: Vista de implementación

el componente de vista de incremento, la suposición de continuidad temporal desempeña un papel fundamental en la identificación de valores atípicos. La continuidad temporal se refiere al hecho de que no se espera que los patrones en los datos cambien abruptamente a menos que haya procesos anormales en el trabajo.

6. Actividades complementarias

6.1 LISS 2010 (9th International Conference on Logistics, Informatics and Service Sciences)

Congreso LISS 2019 celebrado en la Universidad de Maryland del 26 al 29 de Julio de este año. Este evento fue acreditado por el Comité Técnico de IEEE sobre Informática, el Centro Internacional de Investigación Informática de la Universidad Jiaotong de Beijing, en cooperación con la Universidad de Maryland, Estados Unidos y la Universidad de Reading, Reino Unido. En dicho evento participé en la sesión de Análisis de datos y toma de decisiones empresariales con la presentación del artículo de investigación “Trends and Challenges of Data Management in Industry 4.0”. Este artículo de investigación está en proceso de ser incluido en IEEE Xplore y otras bases de datos de indexación.

6.2 CONISOFT 2019 (The 7th International Conference in Software Engineering Research and Innovation)

Congreso internacional CONISOFT 2019 celebrado en la facultad de ingeniería de la UNAM del 23 al 25 de octubre. Evento Técnicamente copatrocinado por la IEEE y por la IEEE Computer Society. En dicho evento participe en el Pista 1: Documentos científicos (investigación original, básica y experimental), Este artículo de investigación está en proceso de ser incluido en IEEE Xplore y otras bases de datos de indexación.

6.3 Artículo para revista JCR

En cumplimiento con el reglamento del doctorado que pide presentar el avance en la redacción del manuscrito de un artículo JCR, se completó el artículo. El artículo en cuestión lleva el título de "*Data management drivers for real-time analytics in industry 4.0: A centralized approach"*. La revista tentativa es IEEE Transactions on Systems, Man, and Cybernetics: Systems, con un factor de impacto de 7.3 y un tiempo de publicación de 24.3 semanas.

6.4 Calendario de actividades

TABLA IX. ACTIVIDADES

No.	Actividad	S1	S2	S3	S4	S5	S6	S7	S8
1	Revisión sistemática de literatura del estado del arte								
2	Primer Artículo para Congreso Nacional								
3	Elaboración del protocolo de investigación								
4	Artículo para congreso internacional								
5	Arquitectura de optimización y predicción								
6	Modelo de detección de anomalías								
7	Elaborar Paper JCR								
8	Arquitectura para analítica en industria 4.0								
9	Elaborar segundo Paper JCR								
10	Modelo de Privacidad								
11	Integrar a la arquitectura de optimización el modelo de privacidad								
12	Desarrollo del prototipo través de un caso de uso								
13	Artículo (congreso Internacional)								
14	Escribir la tesis								
15	Elaborar el tercer Paper JCR (resultado del caso de uso)								
16	Producto bajo esquema de propiedad intelectual								
17	Presentación de defensa de Tesis								

7. Conclusiones

- Se proponen seis escenarios para desarrollar analítica de big data en la industria 4.0.
- Se presenta un modelo de detección de anomalías para datos temporales.
- Se proponen cuatro casos de uso para validar el modelo de detección de anomalía propuesto, además de ser la base para el desarrollo de la metodología de implementación.
- Se desarrolló el caso detección de eventos de fallo en plantas industriales en el escenario de equipos de proceso.
- Se presenta la propuesta para una arquitectura para la analítica de big data en la industria 4.0.
- Dentro de las actividades complementarias se destaca la participación en dos congresos internacionales avalados por la IEEE y la conclusión de un artículo para revista JCR.

Referencias

[1] F. Tao, Q. Qi, A. Liu, and A. Kusiak, "Data-driven smart manufacturing," *J. Manuf. Syst.*, vol. 48, pp. 157–169, Jul. 2018.

[2] Q. Qi and F. Tao, "Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison," *IEEE Access*, vol. 6, pp. 3585–3593, 2018.

[3] Y. Bai, Z. Sun, J. Deng, L. Li, J. Long, and C. Li, "Manufacturing quality prediction using intelligent learning approaches: A comparative study," *Sustain.*, vol. 10, no. 1, p. 85, Dec. 2017.

[4] J. Lee, E. Lapira, B. Bagheri, and H. an Kao, "Recent advances and trends in predictive manufacturing systems in big data environment," *Manuf. Lett.*, vol. 1, no. 1, pp. 38–41, 2013.

[5] J. Li, F. Tao, Y. Cheng, and L. Zhao, "Big Data in product lifecycle management," *Int. J. Adv. Manuf. Technol.*, vol. 81, no. 1–4, pp. 667–684, Oct. 2015.

[6] N. Jesse, "Internet of Things and Big Data: the disruption of the value chain and the rise of new software ecosystems," *AI Soc.*, vol. 33, no. 2, pp. 229–239, 2018.

[7] J. Lee, B. Bagheri, and H.-A. Kao, "Recent Advances and Trends of Cyber-Physical Systems and Big Data Analytics in Industrial Informatics," *Int. Conf. Ind. Informatics*, no. November 2015, pp. 1–6, 2014.

[8] & R. Bagozi, A., Bianchini, D., De Antonellis, V., Marini, A., "Summarisation and Relevance Evaluation Techniques for Big Data Exploration: The Smart Factory Case Study," in *International Conference on Advanced Information Systems Engineering*, 2017, vol. 10253 LNCS, pp. V264-279.

[9] R. Atat, L. Liu, J. Wu, G. Li, C. Ye, and Y. Yang, "Big Data Meet Cyber-Physical Systems: A Panoramic Survey," *IEEE Access*, vol. 6, pp. 73603–73636, 2018.

[10] "Outlier Detection Using Spark Streaming," no. December, 2017.

[11] A. Lazarevic and V. Kumar, "Feature bagging for outlier detection," in *Proceeding of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining - KDD '05*, 2005, p. 157.

[12] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, 2009.

[13] Y. Zhao and M. K. Hryniewicki, "XGBOD: Improving Supervised Outlier Detection with Unsupervised Representation Learning," *Proc. Int. Jt. Conf. Neural Networks*, vol. 2018-July, 2018.

[14] M. Goldstein and S. Uchida, "A comparative evaluation of unsupervised anomaly detection algorithms for multivariate data," *PLoS One*, vol. 11, no. 4, p. e0152173, Apr. 2016.

[15] C. C. Aggarwal, "Outlier analysis," in *Outlier Analysis*, vol. 9781461463, Cham, Switzerland: Springer Nature, 2013, pp. 1–446.

[16] C. Aggarwal, "Outlier ensembles: position paper," *ACM SIGKDD Explor. Newsl.*, vol. 14, no. 2, pp. 49–58, 2013.

[17] C. C. Aggarwal and S. Sathe, "Theoretical Foundations and Algorithms for Outlier Ensembles," *ACM SIGKDD Explor. Newsl.*, vol. 17, no. 1, pp. 24–47, Sep. 2015.

[18] C. Zhang and Y. Ma, *Ensemble machine learning: Methods and applications*. New York, New York, USA: pringer Science & Business Media, 2012.

[19] Z. H. Zhou, *Ensemble methods: foundations and algorithms*. 6000 Broken Sound Parkway NW: Chapman and Hall/CRC, 2012.

[20] Y. Zhao, Z. Nasrullah, and Z. Li, "PyOD: A Python Toolbox for Scalable Outlier Detection," *ournal Mach. Learn. Res.*, vol. 20, no. 96, pp. 1–7, 2019.

[21] E. Achtert, H. P. Kriegel, L. Reichert, E. Schubert, R. Wojdanowski, and A. Zimek, "Visual evaluation of outlier detection models," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 5982 LNCS, no. PART 2, pp. 396–399, 2010.

[22] R. Hofmann, M., & Klinkenberg, *RapidMiner: Data mining use cases and business analytics applications*. CRC Press, 2013.

[23] L. Komsta, "outliers: Tests for outliers. R package version 0.14," *Repository CRAN*, 2011. [Online]. Available: http://www.r-project.org.

[24] M. L. Shyu, S. C. Chen, K. Sarinnapakorn, and L. Chang, "Principal component-based anomaly detection scheme," *Stud. Comput. Intell.*, vol. 9, pp. 311–329, 2006.

[25] J. Hardin and D. M. Rocke, "Outlier detection in the multiple cluster setting using the minimum covariance determinant estimator," *Comput. Stat. Data Anal.*, vol. 44, no. 4, pp. 625–638, 2004.

[26] P. J. Rousseeuw and K. Van Driessen, "A fast algorithm for the minimum covariance determinant estimator," *Technometrics*, vol. 41, no. 3, pp. 212–223, 1999.

[27] B. Schölkopf, J. C. Platt, J. Shawe-Taylor, A. J. Smola, and R. C. Williamson, "Estimating the support of a high-dimensional distribution," *Neural Comput.*, vol. 13, no. 7, pp. 1443–1471, 2001.

[28] J. Breunig, M.M., Kriegel, H.P., Ng, R.T. and Sander, "LOF: Identifying Density-Based Local Outliers," *ACM sigmod Rec.*, vol. 29, no. 2, pp. 93–104, 2000.

[29] J. Tang, Z. Chen, A. W. C. Fu, and D. W. Cheung, "Enhancing effectiveness of Outlier detections for low Density Patterns," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2336, pp. 535–548, 2002.

[30] O. D. Rfdo *et al.*, "Discovering cluster-based local outliers," *Pattern Recognit. Lett.*, vol. 24, no. 9–10, pp. 1641–1650, 2003.

[31] C. Papadimitriou, S., Kitagawa, H., Gibbons, P. B., & Faloutsos, "Loci: Fast outlier detection using the local correlation integral," in *Proceedings 19th International Conference on Data Engineering (Cat. No. 03CH37405)*, 2003, pp. 315–326.

[32] M. Goldstein and A. Dengel, "Histogram-based outlier score (hbos): A fast unsupervised anomaly detection algorithm," *KI-2012 Poster Demo Track*, no. 1, pp. 59–63, 2012.

[33] S. Ramaswamy, R. Rastogi, and K. Shim, "Efficient algorithms for mining outliers from large data sets," *SIGMOD Rec. (ACM Spec. Interes. Gr. Manag. Data)*, vol. 29, no. 2, pp. 427–438, 2000.

[34] F. Angiulli and C. Pizzuti, "Fast outlier detection in high dimensional spaces," *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 2431 LNAI, pp. 15–27, 2002.

[35] H.-P. Kriegel, P. Kroger, E. Schubert, and A. Zimek, "Outlier Detection in Axis-Parallel Subspaces of High Dimensional Data BT - Advances in Knowledge Discovery and Data Mining," *Adv. Knowl. Discov. Data Min.*, vol. 5476, no. Chapter 86, pp. 831–838, 2009.

[36] H. Kriegel and M. Schubert, "Angle-Based Outlier Detection in High-dimensional Data," in *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2008, pp. 444–452.

[37] H. . Janssens, J.H.M., Huszár, F., Postma, E.O. and van den Herik, “Stochastic outlier selection,” Tilburg, The Netherlands., 2012.
[38] C. C. Aggarwal, *Outlier analysis. In Data mining*. Springer, Cham., 2015.
[39] Y. Liu *et al.*, “Generative Adversarial Active Learning for Unsupervised Outlier Detection,” *EEE Trans. Knowl. Data Eng.*, pp. 1–13, 2019.
[40] N. Marz, *Principles and best practices of scalable real-time data systems*, vol. 37. 2015.
[41] M. Gupta, “Outlier Detection for Temporal Data: A Survey,” *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2250–2267, 2013.
[42] M. Tahmassebpour, “A New Method for Time-Series Big Data Effective Storage,” *IEEE Access*, vol. 5, no. c, pp. 10694–10699, 2017.
[43] Y. Zhang, N. Meratnia, and P. Havinga, “Outlier detection techniques for wireless sensor networks: A survey,” *IEEE Commun. Surv. Tutorials*, vol. 12, no. 2, pp. 159–170, 2010.
[44] C. C. Aggarwal and S. Sathe, *Outlier ensembles: An introduction*. 2017.
[45] Prognostics and Health Management Society, “PHM data challenge 2015,” 2015. .
[46] H. Kim *et al.*, “Fault log recovery using an incomplete-data-trained FDA classifier for failure diagnosis of engineered systems,” in *Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM*, 2015, no. 2006, pp. 736–745.
[47] W. Xiao, “A probabilistic machine learning approach to detect industrial plant faults: PHM15 data challenge,” *Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM*. pp. 718–726, 2015.
[48] C. Xie, D. Yang, Y. Huang, and D. Sun, “Feature extraction and ensemble decision tree classifier in plant failure detection,” in *Proceedings of the Annual Conference of the Prognostics and Health Management Society, PHM*, 2015, pp. 727–735.
[49] N. Marz, *Principles and best practices of scalable real-time data systems*, vol. 37. 2015.