Propuesta de Arquitectura de Software para la Detección Automatizada de Anomalías en Procesos de Atornillado

1. Comprensión del Problema y Requisitos

Objetivo del Proyecto:

- Desarrollar un sistema automatizado para detectar anomalías en el proceso de atornillado utilizando aprendizaje automático.

Proceso Actual:

- Los datos en tiempo real de los sensores sobre ángulo y torque se comparan con parámetros preconfigurados para identificar anomalías.

Resultado Deseado:

- Implementar una solución de aprendizaje automático para automatizar la detección de anomalías sin necesidad de preconfiguración, mejorando la precisión.

Restricciones:

- Despliegue local (no es necesario el despliegue en servidor).
- Los usuarios son ingenieros que actualmente monitorean gráficos visualmente.

2. Limpieza y Preparación de Datos

Fuentes de Datos:

- Archivos Excel que contienen datos de ángulo y torque para cada operación de atornillado, etiquetados como OK (0) o NOK (2).

Pasos para la Limpieza de Datos:

1. **Cargar Datos:** Leer los archivos Excel en Python utilizando pandas.
2. **Consistencia de Datos:** Asegurar que el formato de los datos sea consistente en todos los archivos.
3. **Manejo de Valores Faltantes:** Identificar y manejar cualquier entrada de datos faltante o inconsistente.
4. **Ingeniería de Características:** Crear características adicionales si es necesario, como estadísticas resumidas (media, varianza) de ángulo y torque.

3. Análisis de Datos y Construcción del Modelo

Análisis Exploratorio de Datos (EDA):

- Visualizar datos para entender distribuciones e identificar patrones o anomalías.
- Utilizar matplotlib o seaborn para graficar.

Modelos de Aprendizaje Automático:

- **Selección de Modelos:** Basado en la naturaleza del problema, los modelos adecuados incluyen:
  - **Random Forest Classifier:** Por su robustez y capacidad para manejar datos desequilibrados.
  - **Support Vector Machine (SVM):** Por su efectividad en espacios de alta dimensionalidad.
  - **Redes Neuronales:** Para capturar patrones complejos en los datos.

Pasos para la Construcción del Modelo:

1. **División de Datos:** Dividir los datos en conjuntos de entrenamiento y prueba.
2. **Entrenamiento del Modelo:** Entrenar los modelos seleccionados en los datos de entrenamiento.
3. **Optimización de hiperparámetros**: Realizar exploración de hiperparámetros para encontrar el mejor modelo posible.

4. **Evaluación del Modelo:** Evaluar los modelos utilizando métricas como precisión, precisión, recall y F1-score.

4. Arquitectura del Software

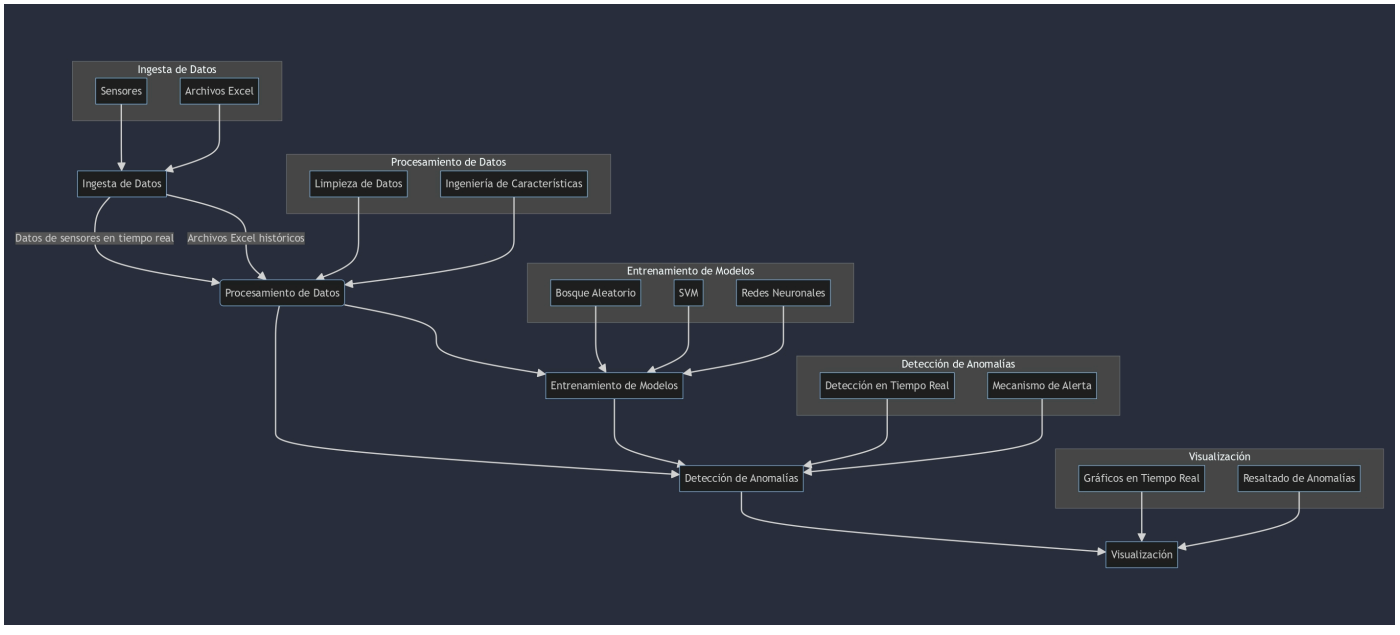
Resumen:

- El sistema consta de varios módulos, incluyendo ingesta de datos, procesamiento de datos, entrenamiento de modelos, detección de anomalías y visualización.

Módulos:

1. **Ingesta de Datos:**
    - Cargar datos en tiempo real desde sensores y datos históricos desde archivos Excel.
2. **Procesamiento de Datos:**
    - Limpiar y preprocesar los datos, realizar ingeniería de características.
3. **Entrenamiento de Modelos:**
    - Entrenar modelos de aprendizaje automático en datos históricos, actualizar modelos periódicamente.
4. **Detección de Anomalías:**
    - Utilizar modelos entrenados para detectar anomalías en datos en tiempo real.
5. **Visualización:**
    - Generar gráficos en tiempo real para ángulo y torque, resaltar anomalías detectadas.

Diagrama de Arquitectura:



Explicación Detallada:

1. **Ingesta de Datos:**
    - Extraer datos periódicamente desde sensores y archivos Excel.
    - Utilizar pandas para la manipulación y carga de datos.
2. **Procesamiento de Datos:**
    - Limpiar y preprocesar datos para asegurar consistencia.
    - Realizar ingeniería de características para extraer características útiles.
3. **Entrenamiento de Modelos:**
    - Entrenar múltiples modelos para comparar su rendimiento.
    - Utilizar validación cruzada para asegurar robustez.
4. **Detección de Anomalías:**
    - Desplegar el modelo de mejor rendimiento para la detección de anomalías en tiempo real.
    - Configurar mecanismos de alerta para anomalías detectadas.
5. **Visualización:**
    - Utilizar matplotlib o Plotly para crear visualizaciones interactivas.
    - Resaltar anomalías en los gráficos para facilitar el monitoreo.

5. Conclusión

Esta propuesta de arquitectura de software detalla los pasos y módulos necesarios para desarrollar un sistema automatizado de detección de anomalías en procesos de atornillado. Siguiendo este plan, se puede asegurar una solución robusta y eficiente que cumpla con los objetivos y restricciones del proyecto.