**Relatório de Análise do Código: Detecção de Anomalias em Motor de Vibração**

**Introdução**

O código analisado implementa um pipeline completo para a detecção de anomalias em dados de corrente de um motor de vibração, utilizando aprendizado profundo. O objetivo principal é identificar padrões anômalos nas fases de corrente (A, B e C) a partir de dados temporais.

**1. Configuração Inicial**

O código inicia com a instalação das bibliotecas essenciais: scikit-learn, pandas, numpy, matplotlib, tensorflow e shap. A verificação da GPU é realizada para otimizar o treinamento do modelo.

**2. Carregamento e Pré-processamento dos Dados**

**Parâmetros de Configuração:**

* WINDOW\_SIZE: 60 amostras (equivalente a 1 segundo de dados).
* STEP\_SIZE: 30 amostras (50% de sobreposição entre janelas).

**Função de Janelamento**

A função create\_sequences é responsável por criar sequências temporais a partir dos dados. Ela verifica a consistência temporal e ignora janelas com gaps maiores que 100ms.

**3. Leitura dos Dados**

Os dados são carregados de arquivos CSV representando dados saudáveis (healthy.csv) e dados com falha (0.7inner-200watt.csv).

**Padronização dos Nomes das Colunas**

Uma função clean\_column\_name é utilizada para remover espaços, hífens e padronizar a nomenclatura. Além disso, a coluna de timestamp é renomeada para timestamp.

**Rotulagem dos Dados**

* healthy['label'] = 0
* faulty['label'] = 1

**4. Engenharia de Features**

A função calculate\_phase\_features calcula estatísticas adicionais, como:

* current\_avg: Média das correntes das fases A, B e C.
* imbalance: Desvio padrão das fases.
* max\_phase e min\_phase: Máximos e mínimos das fases.
* thd: Distorção harmônica total.

**5. Criação das Janelas Temporais**

A função create\_sequences gera janelas deslizantes de dados temporais, considerando oito características extraídas de cada amostra.

**6. Definição do Modelo**

**Arquitetura do Modelo:**

* Camada convolucional Conv1D para extração de características.
* Camada Bidirectional LSTM para captura de padrões temporais.
* Camadas de combinação Concatenate.
* Camada final densa Dense com ativação sigmoide para classificação binária.

O modelo é compilado com o otimizador Adam e a função de perda binary\_crossentropy.

**7. Divisão dos Dados**

Os dados são divididos temporalmente em 80% para treinamento e 20% para teste.

**8. Treinamento do Modelo**

O treinamento ocorre com callbacks para early stopping e salvamento do melhor modelo com base na métrica AUC.

**9. Avaliação do Modelo**

O modelo salvo é carregado e avaliado em diferentes conjuntos de dados de teste (de healthy.csv a 1.7inner-200watt.csv). Anomalias são detectadas com base em um limiar de 0.5.

**Conclusão**

O código apresenta um pipeline robusto para a detecção de anomalias em motores de vibração, combinando engenharia de features, aprendizado profundo e avaliação contínua. A abordagem com janelas deslizantes permite capturar padrões temporais, enquanto a arquitetura CNN-LSTM melhora a capacidade de generalização.