UNIVERSIDADE FEDERAL DE ALAGOAS-UFAL
CENTRO DE TECNOLOGIA CTEC
CURSO DE ENGENHARIA QUÍMICA

ELBER CAIO ANTHONY CADETE LOPES

APLICAÇÃO DE TÉCNICA DE ANÁLISE MULTIVARIADA EM CONJUNÇÃO DE TÉCNICAS DE APRENDIZADO DE MÁQUINA E APRENDIZADO PROFUNDO, EMPREGADOS PARA DETECÇÃO E DIAGNOSTICO DE FALHAS NA INDÚSTRIA.

MACEIÓ
2022

Elber Caio Anthony Cadete Lopes

APLICAÇÃO DE TÉCNICA DE ANÁLISE MULTIVARIADA EM CONJUNÇÃO DE TÉCNICAS DE APRENDIZADO DE MÁQUINA E APRENDIZADO PROFUNDO, EMPREGADOS PARA DETECÇÃO E DIAGNOSTICO DE FALHAS NA INDÚSTRIA.

Monografia apresentada como requisito parcial para obtenção do grau de Bacharel em Curso de Engenharia Química da Universidade Federal de Alagoas - UFAL, Centro de Tecnologia – CTEC.

Orientador: Prof. Dr. Frede de Oliveira Carvalho

Maceió
2022

Catalogação na Fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 – 1767

L864a Lopes, Elber Caio Anthony Cadete.

Aplicação de técnica de análise multivariada em conjunção de técnicas de aprendizado de máquina e aprendizado profundo, empregados para detecção e diagnóstico de falhas na indústria / Elber Caio Anthony Cadete Lopes. – 2022.

55 f. il. : figs. ; tabs. color.

Orientador: Frede de Oliveira Carvalho.

Monografia (Trabalho de Conclusão de Curso em Engenharia Química). Universidade Federal de Alagoas. Centro de Tecnologia. Maceió, 2022.

Bibliografia: f. 44-45.

Apêndices: f. 46-55.

1. Análise multivariada. 2. Aprendizagem de máquina. 3. Aprendizado profundo. I. Título.

CDU: 004.81:159.953.5

Dedico esse trabalho à minha mãe pela contribuição na formação do meu caráter. Obrigado por ter fé em mim e orgulho da minha trajetória.

AGRADECIMENTOS

Gostaria de agradecer e dedicar este trabalho às seguintes pessoas:

Minha família, minha mãe Elisângela, o pilar que me apoia para tudo, minha avó Eliene e meu pai Edmilson que estiveram sempre ao meu lado.

Meus amigos de infância, Aguinaldo, Arnon, Jâmisson e João, insubstituíveis em minha jornada.

Aos meus amigos e colegas da faculdade, com quem compartilhei vários momentos. Em especial à minha turma: Aline, Caju, Marquinhos, Rafael e Solleti, e ao meu amigo, Lucas que foi essencial na confecção deste trabalho e em me manter focado na graduação.

E por último ao meu orientador Frede e ao LABSIA, pelo incentivo, paciência e dedicação, contribuindo não só para este trabalho, mas para minha vida acadêmica e profissional.

O passo mais importante que um homem pode dar. Não é o primeiro, é? É o próximo. Sempre o próximo passo.

Brandon Sanderson

RESUMO

A estabilidade e segurança de uma indústria química dependem de sua capacidade de monitoramento para Detecção e Diagnóstico de Falhas (FDD), definidas como comportamentos indesejados de parâmetros do processo. A detecção das falhas e identificação de suas causas fornecem uma base de decisão para manutenção de equipamentos, assegurando a capacidade de operação de processos químicos de maneira segura, garantindo proteção de pessoal e de equipamento. Uma linha de produção moderna pode providenciar a quantidade necessária de dados para resolver esse problema, porém, tendo em vista a complexidade dos processos industriais, a utilização de métodos mais tradicionais como a modelagem fenomenológica e a análise de sinais se tornam muito difíceis, abrindo espaço para a utilização de técnicas de análise multivariada, como a Análise de Componentes Principais (PCA), em conjunto com técnicas de aprendizado de máquina, ou de técnicas de aprendizado profundo, sendo estas duas últimas pertencentes ao contexto da indústria 4.0, dependentes de *Big Data*. Isto posto, este trabalho teve como objetivo desenvolver, para uma aplicação industrial, estratégias, na forma de rotinas computacionais, construídas com a linguagem de programação *Python* para detecção de falhas com PCA e do diagnóstico das falhas detectadas tanto com o aprendizado de máquina, utilizando *Random Forest* (RF) quanto com o aprendizado profundo através das redes neurais recorrentes (RNN) do tipo *Gated Recurrent Unit* (GRU), além disso, para a validação das rotinas computacionais desenvolvidas foi utilizado a simulação do processo químico *Tennessee Eastman Process* (TEP) que é muito utilizado como *benchmark* em estudos de controle, modelagem e tratamento de dados, por ter um comportamento próximo ao de uma planta real completa e ser um problema complexo. O PCA apresentou bons resultados na detecção apenas para 11 das 20 falhas, a RF conseguiu diagnosticar eficientemente 14 das 20 falhas, já a GRU superou bastante a performance da RF para todas as falhas, mesmo que para isso foi preciso o dobro de memória RAM e mais de 10 vezes o tempo de execução.

Palavras-chave: Técnica de análise multivariada. Detecção e diagnostico de falhas. Aprendizado de máquina e Aprendizado profundo.

ABSTRACT

The stability and safety of chemical industries depends on its monitoring capacity for Fault Detection and Diagnosis (FDD), faults are defined as unwanted behaviors of process parameters. The detection of failures and identification of their causes provide a basis for decision-making on equipment maintenance, which ensures the ability to operate chemical processes safely, guaranteeing protection of personnel and equipment. A modern production line can provide the necessary amount of data to solve this problem, however, given the complexity of industrial processes, the use of more traditional methods such as phenomenological modeling and signal analysis become very difficult, making space for the use of multivariate analysis techniques, such as Principal Component Analysis (PCA), in conjunction with machine learning techniques, or deep learning techniques, the latter two belonging to the context of the 4.0 industry, dependent on Big data. That said, this work aimed to develop, for an industrial application, strategies, in the form of computational routines, built with the programming language Python for fault detection with PCA and the fault diagnosis using machine learning through the Random Forest (RF) technique and deep learning through Recurrent Neural Networks (RNN) specifically the Gated Recurrent Unit (GRU), in addition, for the validation of the developed computational routines, The simulation of the chemical process Tennessee Eastman Process (TEP) was chosen, which is widely used as a benchmark in studies of control, modeling and data processing, as it behaves close to that of a complete real plant and is a complex problem. The PCA showed good results in detecting 11 of the 20 faults, the RF was able to efficiently diagnose 14 of the 20 faults, while the GRU far surpassed the RF performance for all failures, even though it took twice as much RAM and more than 10 times the execution time.

Keywords: Multivariate analysis technique. Fault detection and diagnosis. Machine Learning and Deep Learning.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

FDD	*Fault Detection and Diagnosis*
PCA	*Principal Component Analisys*
RF	*Random Forest*
RNN	*Recurrent Neural Network*
GRU	*Gated Recurrent Unit*
TEP	*Tennessee Eastman Process*
LABSIA	Laboratório de Sistemas Inteligentes Aplicados
TP	Verdadeiros Positivos
TN	Verdadeiros Negativos
FP	Falsos Positivos
FN	Falsos Negativos
ROC	*Receiver Operating Characteristic*
AUC	Área Sob a Curva ROC

SUMÁRIO

1 INTRODUÇÃO

O cenário industrial é permeado pela necessidade de se fazer concessões para seguir diversas obrigações, primariamente para garantir a qualidade do produto final, gerando a necessidade de gerenciar o consumo de energia e material, mas também para assegurar a segurança dos funcionários e seguir as regulamentações ambientais, para isso é necessário um profundo conhecimento quantitativo do processo e de seus parâmetros relevantes (FORTUNA et al., 2007).

Esse conhecimento é possibilitado pelos avanços tecnológicos e científicos que geram grandes mudanças nos processos químicos e permitem que a indústria química se torne mais inteligente e automatizada (SHAO et al., 2019).

Com a chamada revolução industrial 4.0, existe uma tendência do aumento da comunicação entre os diversos setores e equipamentos na indústria, que se manifesta com o barateamento de sensores e estocagem de dados, aumento da instrumentação disponível e da melhora na comunicação entre servidores, plantas e operadores, resultando em um aumento enorme na quantidade de dados gerados e em um dos pilares da indústria 4.0, o Big Data (SOARES, 2017).

Com o aumento da complexidade dos processos industriais e por conta de seu potencial risco, uma das análises necessárias dos dados gerados na indústria é a Detecção e Diagnóstico de Falhas, do inglês *Fault Detection and Diagnosis* (FDD), onde falha é qualquer estado sistemático que se afasta do normal. Suas causas são inúmeras, e podem ser inconsequentes ou comprometer o sistema inteiro, o que pode ser evitado com a FDD, detectando e diagnosticando a falha, por técnicas de classificação, a tempo de o operador tomar as ações necessárias, aumentando produtividade, diminuindo custo de manutenção e melhorando a utilização do processo da produção (DRAGOGIAS, 2019).

Entre as múltiplas técnicas utilizadas para detecção de falha, uma das mais utilizadas é a Análise de Componentes Principais, do inglês *Principal Component Analisys* (PCA), uma técnica linear multivariada de redução de dimensionalidade que permite analisar a variabilidade dos dados da planta química para estabelecer um modelo do estado normal do processo (CHIANG et al., 2001). Isso possibilita a utilização das estatísticas T^2 e Q, definindo como falha todos os estados em que uma dessas estatísticas sai de seus limites (SORAYA et al., 2019).

Para o diagnóstico das falhas detectadas, se destaca o aprendizado de máquina, um dos adventos da indústria 4.0, resultado da evolução da detecção de padrões e da teoria de aprendizado computacional, que com sua capacidade treinar uma máquina para prever valores ou classificar dados consegue discernir qual o tipo da falha ocorreu no sistema.

Porém, com o aumento da quantidade de dados, inevitável na era do Big Data, o treinamento se torna mais custoso em questão de tempo e processamento e a precisão dos modelos diminui, mas com a evolução rápida do aprendizado de máquina, veio o aprendizado profundo, desenvolvido de maneira a contornar esses inconvenientes de maneira eficiente, que pode analisar dados brutos em um nível maior de abstração através da combinação de uma série de funções não lineares, para poder processar efetivamente os dados de processos complexos (YUAN; TIAN, 2019).

Durante a graduação foram desenvolvidos por este autor, no âmbito do Laboratório de Sistemas Inteligentes Aplicados (LABSIA), diversos trabalhos abordando diversas técnicas de aprendizado de máquina de classificação para diagnóstico de falhas, como as Redes Neurais Artificiais, as Máquinas de Vetores Suporte e até mesmo o PCA, escolhendo-se dessa forma com um caráter de expandir os conhecimentos, avaliar um modelo de aprendizado de máquina comum na indústria, as *Random Forests* (RFs) e para aplicar as inovações do aprendizado profundo na indústria química, foi utilizado a *Gated Recurrent Unit* (GRU).

Um dos algoritmos mais utilizados do aprendizado de máquina são as Florestas Aleatórias, do inglês, *Random Forests*, modelos não lineares de regressão ou classificação, que consistem em conjuntos (*Ensemble*) de árvores de decisão que participam de um processo de voto para classificar ou prever dados, incluindo a capacidade de classificar categorias de falhas (YANG et al., 2008).

Além do ineditismo e da popularidade, a RF também aparenta ter melhor resultado para certos processos químicos, superando outros classificadores convencionais como Redes Neurais Artificiais, Máquinas de Vetores Suporte, *K-Nearest Neighbors* e Árvores de Decisões simples, sendo outros bons candidatos de técnicas para o FDD (SHAO et al., 2019).

Nas técnicas de aprendizado profundo, por levar completamente em conta as associações entre as amostras, as Redes Neurais Recorrentes, do inglês, *Recurrent Neural Network* (RNN) é uma das mais adequadas no diagnóstico de falhas de sistemas dinâmicos, sendo uma técnica bem robusta pela maneira em que ocorrem as conexões dos neurônios na camada oculta.

No entanto, as RNNs apresentam problemas no treinamento, o que gerou a necessidade de variantes, principalmente as que adicionam operações de portões (*gates*), pois, além de reterem todas as vantagens da RNN, usam portões nos neurônios de camada oculta que permitem manter informações úteis e descartar informações inúteis em dados de sequência dinâmica de maneira automática. A variante de RNN mais popular é a *Long short-term memory*, mas a constante inovação da indústria levou a outras alternativas, como a GRU que otimiza o uso dos portões

(YUAN; TIAN, 2019).

Computacionalmente, para implementação do sistema de FDD, a linguagem de programação *Python* parece ser ideal, graças a sua extensa quantidade de bibliotecas de livre acesso e ser bem eficiente para as aplicações do trabalho, tomando cada vez mais espaço na análise de dados de processos químicos (CAVALCANTI et al., 2021).

Desta maneira, neste trabalho será realizado o desenvolvimento e análise de estratégias computacionais utilizando a linguagem *Python* para aplicação da detecção de falhas utilizando PCA e do diagnóstico de falhas através de RF e GRU, as quais receberão uma avaliação comparativa de eficiência. Vale salientar que para a validação das rotinas computacionais desenvolvidas foi utilizado a simulação do processo químico *Tennessee Eastman Process* (TEP) que é muito utilizado como *benchmark* em estudos de controle, modelagem e tratamento de dados, por ter um comportamento próximo ao de uma planta real completa e ser um problema complexo, com dados suficientes para se caracterizar como Big Data.

2 OBJETIVOS

2.1 GERAIS

Desenvolvimento e avaliação de estratégias na linguagem de programação *Python* para implementação de um sistema de detecção de falhas com técnica de análise multivariada e diagnóstico das falhas detectadas tanto com o aprendizado de máquina, quanto com o aprendizado profundo aplicados ao processo químico (TEP).

2.2 ESPECÍFICOS

- Avaliar a aplicabilidade da Análise de PCA, como técnica de detecção de falhas no TEP em linguagem *Python*;
- Avaliar a aplicabilidade da Análise de PCA, como técnica de detecção de falhas no TEP em linguagem *Python*;
- Avaliar a aplicabilidade das redes neurais recorrentes do tipo GRU, como técnica de diagnóstico de falhas no TEP em linguagem *Python*;
- Verificar a viabilidade da técnica de aprendizado profundo em comparação com a de aprendizado de máquina no TEP.

3 REVISÃO BIBLIOGRÁFICA

3.1 ANÁLISE DE COMPONENTES PRINCIPAIS

O PCA, é uma técnica linear de redução da dimensionalidade, que constrói uma representação de baixa dimensionalidade de maneira otimizada para descrever o máximo da variância dos dados, determinando um conjunto de vetores ortogonais, chamados vetores de carregamento (*loadings*), ordenados pela quantidade de variância explicada nas direções do vetor. Dado um conjunto de treinamento de n observações e m variáveis de processo em uma matriz X, os vetores de carregamento são calculados ao decompor X através da decomposição apresentada na Equação (1) (CHIANG et al., 2001):

$$\frac{1}{\sqrt{n-1}}X = U\Sigma V^T \tag{1}$$

Onde $U \in R^{nxn}$ e $V \in R^{mxm}$ são matrizes unitárias e a matriz $\Sigma \in R^{nxm}$ contém os valores reais não negativos da unidade classificados em ordem descendente ao longo da diagonal principal. Para capturar de forma ideal as variações dos dados, minimizando o efeito do ruído aleatório corrompendo a representação do PCA, os vetores de carregamento correspondente a maior valor singular, a, são tipicamente retidos. As observações em X podem ser então projetadas para a matriz score de dimensão inferior T, dada como $T = XP$, onde as colunas da matriz de carregamento $P \in R^{mxa}$, correspondem aos vetores de carregamento associados com o primeiro valor singular a. (CHIANG et al., 2001)

Uma das maneiras de se detectar as condições anormais é através da estatística de *Hotelling* T^2, que pode ser interpretada como uma medição sistemática das variações sistemáticas do processo, e a violação de seu limite indicaria que as variações sistemáticas estão fora de controle. A estatística T^2 pode ser calculada através da Equação (2) (CHIANG et al., 2001):

$$T^2 = x^T P \Sigma_a^{-2} P^T x \tag{2}$$

Onde a contêm as primeiras a linhas e colunas. Já o limite pode ser definido pela Equação (3):

$$T_a^2 = \frac{[(n-1)(n+1)a]}{[n(n-a)]} \times [F(a, n-a)] \tag{3}$$

Com valor de distribuição-F Fa(a, n-a), um nível de significância a e graus de liberdade de a até (n-a).

Vale denotar que a estatística T² é excessivamente sensível às imprecisões no espaço do PCA correspondente aos valores singulares menores, pois mede diretamente a variação ao longo de cada um dos vetores de carregamento. Em outras palavras, mede diretamente os *scores* correspondentes aos valores singulares menores, então a porção do espaço de observação correspondente aos m — a menores valores singulares pode ser monitorada de forma mais robusta usando a estatística Q, também conhecida como *squared prediction error* (SPE), que por não medir as variações ao longo de cada vetor carregamento diretamente, mas sim as somas das variações no espaço residual, não sofre dessa sensibilidade excessiva. A estatística Q pode ser definida na Equação (4) (CHIANG et al., 2001):

$$Q = r^T r, r = (1 - PP^T)x \tag{4}$$

Onde r é o vetor residual, uma projeção de x no espaço residual. De maneira análoga à T², existe um limite que define a condição de anormalidade, que pode ser definido pela Equação (5) (CHIANG et al., 2001):

$$Q_a = \theta_1 \left[\frac{h_0 c_a \sqrt{2\theta_2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right]^{\frac{1}{h_0}} \tag{5}$$

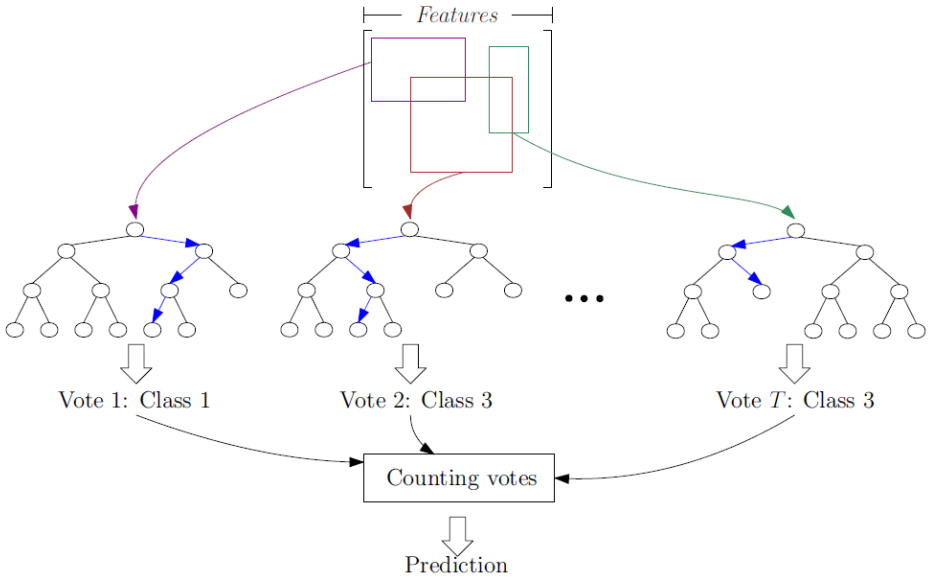
Sendo assim para este trabalho, os dados serão decompostos pelo PCA, serão estimados os limites T^2 e Q, e será calculado o T^2 e Q para cada dado, permitindo analisar quais deles passam deste limite e ainda analisar a contribuição de cada variável para as falhas.

3.2 RANDOM FOREST

Uma das técnicas mais populares de aprendizado de máquina na engenharia é o algoritmo das RFs, um conjunto de árvores de decisão, conhecidos como aprendizes fracos, por ter baixo custo computacional e baixos recursos discriminatórios. Treinar uma RF para classificação é equivalente a treinar várias árvores de decisão independentes, usando a ideia de sabedoria das massas, em que subconjuntos distintos da entrada de dados e recursos extraídos são sorteados aleatoriamente, de modo que cada árvore aprende a partir de uma partição diferente dos dados, e após o procedimento de aprendizagem, uma nova amostra de entrada é rotulada como a classe que a maioria dos classificadores de árvores de decisão votou como descrito na Figura 1 (MARINS et al., 2021).

Fonte: MARINS et al., 2021.

Figura 1 – Diagrama de um classificador por RF.



A distribuição de votos de todas as árvores dentro de uma RF pode ser interpretada como uma distribuição probabilística para a saída do sistema. Esta propriedade pode ser usada para estabelecer um limite para cada classe de saída, com uma amostra dada apenas pertencente a essa classe se a certeza está acima do limite correspondente. No contexto dos sistemas de tomada de decisão, esses limites de classificação permitem equilibrar o número de falsos positivos e falhas não detectadas conforme a prioridade do sistema: aumentar o limite reduz o número de classificações erradas de falhas ao custo de uma redução nas taxas de detecção de falhas (MARINS et al., 2021).

Diversos parâmetros influenciam o desempenho da RF, incluindo o número de árvores construídas, que, em geral, o desempenho aumenta com o aumento do número de árvores, mas também aumenta o tempo computacional, o número mínimo de folhas que devem existir em uma dada árvore, que especifica a quantidade mínima de amostras que um nó deve conter após ser dividido, a profundidade máxima da árvore, que rege a altura máxima até a qual as árvores na floresta podem crescer e o máximo de amostras usados para treinar cada árvore individual (BUITINCK et al., 2013).

Desta forma a RF será aplicada neste trabalho para ilustrar um resultado típico de aprendizado de máquina na classificação das falhas de um processo químico complexo.

3.3 GATED RECURRENT UNIT

As RNNs são chamadas recorrentes, pois executam a mesma atividade para cada elemento em sua sequência, utilizando um estado oculto para registrar o estado momento a momento, enquanto processa a sequência de dados, elas são consideradas parte do aprendizado profundo por possuírem uma estrutura de rede profunda capaz de processar uma quantidade massiva de dados. O modelo é parametrizado através de 3 matrizes, a matriz de pesos W, para as conexões entre a entrada e a camada oculta com o vetor de *bias* b_0, a matriz de pesos R, para as conexões recorrentes nas unidades ocultas e a matriz de pesos V, para as conexões entre a camada oculta e a saída com o vetor de *bias* b_1 (ALIABADI, 2020).

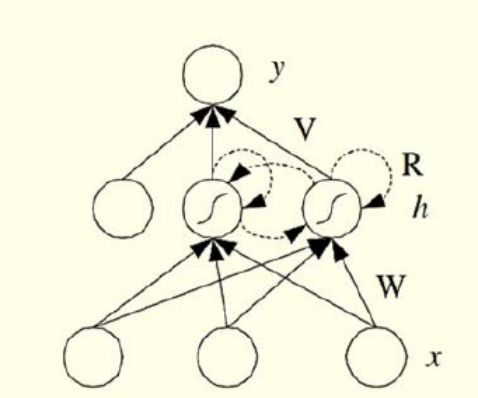
No tempo t, os estados ocultos recebem a soma da entrada com os pesos, x_t e o estado oculto com pesos, h_{t-1} , e passa para a camada de saída após o mapeamento não-linear f, descrito conforme as Equações (6) e (7):

$$h_t = f(h_{t-1}, x_t, \theta) \tag{6}$$

$$\hat{y}_t = g(h_t, \theta) \tag{7}$$

Onde θ são os parâmetros a serem definidos no processo de treinamento, f é a função de ativação não-linear, g é uma função linear para problemas de regressão ou uma função *sigmoid* seguida por uma operação *SoftMax* para problemas de classificação. A arquitetura de uma RNN pode ser constatada na Figura 2.

Figura 2 – A arquitetura de uma RNN.



Fonte: YUAN e YING , 2019.

Embora a RNN seja muito eficiente, existem problemas no treinamento, principalmente quando o tamanho da informação de entrada se torna muito grande, por isso, variantes de RNN foram desenvolvidas para resolver esses problemas, como a *Echo state network*, a *Long short-term memory*, uma das mais antigas (HOCHREITER; SCHMIDHUBER, 1997), e a GRU, proposta por (CHO et al., 2014), que se destaca pelos seus portões (*gates*) evita o chamado *overfitting*, além de economizar tempo de treinamento, que define todos os parâmetros dos pesos, tendo uma maior eficiência, sendo o modelo de RNN mais efetivo para aplicações práticas (YUAN; TIAN, 2019).

Uma representação gráfica da GRU, pode ser observada na Figura 3, ela contém 2 portões, o portão de *reset* r_t e o portão de atualização z_t para atualizar seu estado. O portão de atualização é computado através do estado oculto anterior h_{t-1} e da entrada x_t, visto na Equação (8) (ALIABADI, 2020):

$$z_t = \sigma(W_z x_t + U_z h_{t-1}) \tag{8}$$

Onde σ é a função *sigmoid* logística. O portão *reset* r_t é computado de maneira análoga pela Equação (9):

$$r_t = \sigma(W_r x_t + U_r h_{t-1}) \tag{9}$$

Então é armazenado na memória o $\hat{h_t}$, consoante a Equação (10):

$$\hat{h_t} = \tanh\left(W x_t + r_t \bigodot U h_{t-1}\right) \tag{10}$$

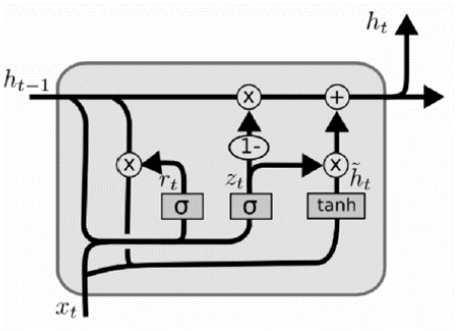
Onde $\bigodot$ é a multiplicação ponto a ponto. Então o estado da célula é atualizado baseado no $\hat{h_t}$ e no estado da célula no tempo t-1 pela Equação (11):

$$h_t = z_t \hat{h_t} + (1 - z_t) h_{t-1} \tag{11}$$

Fonte: ALIABADI, 2020.

Isto posto, neste trabalho, a GRU foi implementada na classificação das falhas do processo, como parte do diagnóstico de falhas em contraponto à RF, sendo uma técnica mais atual e inovadora.

Figura 3 – Representação gráfica de uma GRU.

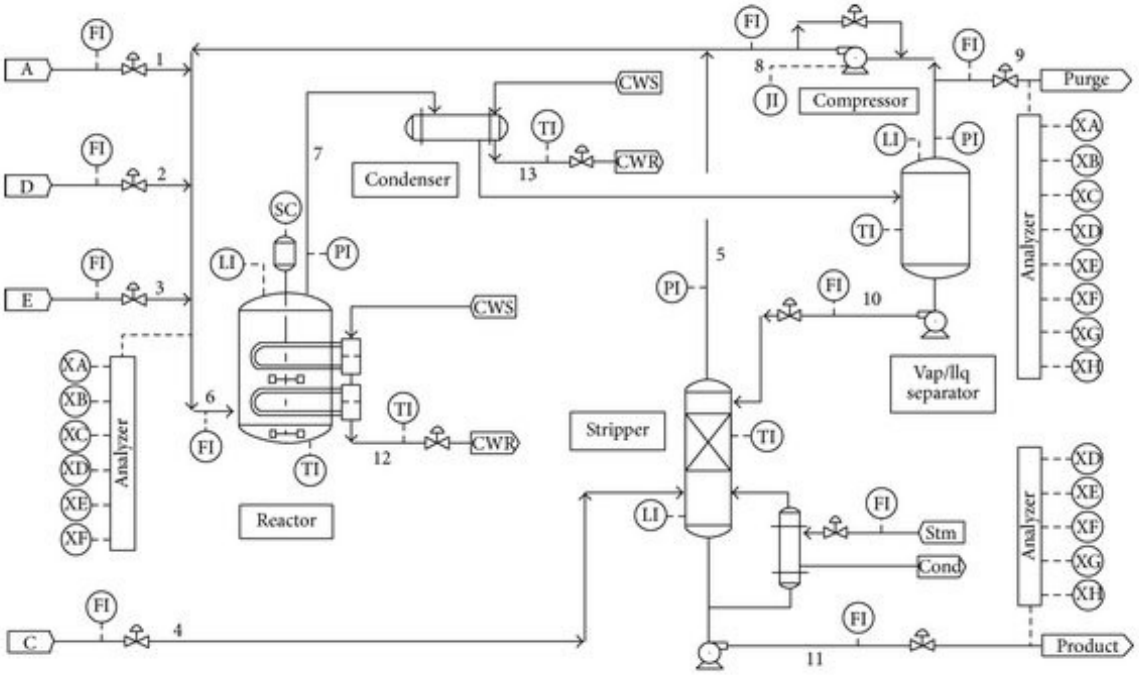


3.4 TENNESSEE EASTMAN PROCESS

A FDD baseada em técnicas de análise multivariada, aprendizado de máquina e aprendizado profundo será aplicada ao TEP, uma planta ilustrada na Figura 4, criada por Downs e Vogel da *Eastman Chemical Company*, de um processo real que foi levemente alterado para proteger a natureza proprietária do processo, especialmente os reagentes, visando fornecer um padrão de simulação de processos complexos de manufatura real. Ele é composto de 5 partes principais: um reator, um condensador, um separador vapor-líquido, um *stripper* para o produto e um compressor para o reciclo. O processo utiliza os reagentes gasosos A, C, D e, além do inerte B, para gerar os produtos G e H, além do subproduto F, conforme as seguintes reações irreversíveis e exotérmicas, com taxas de reação dependentes da temperatura (SORAYA et al., 2019):

$$A_{(g)} + C_{(g)} + D_{(g)} \rightarrow G_{(liq)}, \text{ produto 1;}$$
$$A_{(g)} + C_{(g)} + E_{(g)} \rightarrow H_{(liq)}, \text{ produto 2;}$$
$$A_{(g)} + E_{(g)} \rightarrow F_{(liq)}, \text{ produto 1;}$$
$$3D_{(g)} \rightarrow 2F_{(liq)}, \text{ produto 1;}$$

Figura 4 – Planta do TEP.



Fonte: JIAO et al., 2014.

Os três elementos de natureza gasosa: A, D e E são introduzidos imediatamente no reator, já a espécie C com uma quantidade de A entra na planta (fluxo 4) através do *stripper*. Em seguida, o fluxo de produtos criado após as reações sai do reator e atravessa o condensador, onde os elementos vaporosos se liquefazem, depois, o fluxo do produto é inserido no separador vapor-líquido, enquanto os elementos não condensados são enviados de volta através do compressor para a alimentação do reator. Na coluna de separação a jusante, o resto dos reagentes são eliminados utilizando os elementos da corrente de alimentação 4 como agentes de separação (SORAYA et al., 2019).

Os dados do TEP são compostos por 52 variáveis, consistindo de 11 variáveis manipuladas (XMV), que representam as 11 válvulas controladas (Tabela 1). e 41 variáveis medidas (XMEAS), com 22 medidas do processo envolvendo temperatura, nível, pressão e vazão, (Tabela 2), apresentados, e 19 medições de composição (Tabela 3).

Tabela 1 – Válvulas Controladas.

Variável	Descrição
XMV(1)	Fluxo de alimentação de D (corrente 2)
XMV(2)	Fluxo de alimentação de E (corrente 3)
XMV(3)	Fluxo de alimentação de A (corrente 1)
XMV(4)	Fluxo total de alimentação (corrente 4)
XMV(5)	Válvula de reciclo do compressor
XMV(6)	Válvula de purga (corrente 9)
XMV(7)	Fluxo de líquido do separador (corrente 10)
XMV(8)	Fluxo de líquido do stripper (corrente 11)
XMV(9)	Fluxo de vapor do stripper
XMV(10)	Fluxo de água de resfriamento do reator
XMV(11)	Fluxo de água de resfriamento do condensador

Fonte: SORAYA et al., 2019.

Tabela 2 – Medidas de Processo.

Variável	Descrição	Unidade
XMEAS(1)	Alimentação de A (corrente 1)	kg.m³/h
XMEAS(2)	Alimentação de D (corrente 2)	kg/h
XMEAS(3)	Alimentação de E (corrente 3)	kg/h
XMEAS(4)	Alimentação Total (corrente 4)	kg.m³/h
XMEAS(5)	Fluxo de reciclo (corrente 8)	kg.m³/h
XMEAS(6)	Taxa de alimentação do reator (corrente 6)	kg.m³/h
XMEAS(7)	Pressão do reator	kPa (manométrica)
XMEAS(8)	Nível do reator	%
XMEAS(9)	Temperatura do reator	°C
XMEAS(10)	Taxa de purga (corrente 9)	kg.m³/h
XMEAS(11)	Temperatura de produto do separador	°C
XMEAS(12)	Nível de produto do separador	%
XMEAS(13)	Pressão de produto Sep	kPa (manométrica)
XMEAS(14)	Fluxo de líquido do separador (corrente 10)	m³/h
XMEAS(15)	Nível do stripper	%
XMEAS(16)	Pressão do stripper	kPa (manométrica)
XMEAS(17)	Fluxo de líquido do stripper (corrente 11)	m³/h
XMEAS(18)	Temperatura do stripper	°C
XMEAS(19)	Fluxo de vapor do stripper	kg/h
XMEAS(20)	Trabalho do compressor	kW
XMEAS(21)	Temperatura de saída de água de resfriamento do reator	°C
XMEAS(22)	Temperatura de saída de água de resfriamento do separador	°C

Fonte: SORAYA et al., 2019.

Tabela 3 – Medidas de Composição.

Variável	Descrição	Corrente
XMEAS(23)	Componente A	6
XMEAS(24)	Componente B	6
XMEAS(25)	Componente C	6
XMEAS(26)	Componente D	6
XMEAS(27)	Componente E	6
XMEAS(28)	Componente F	6
XMEAS(29)	Componente A	9
XMEAS(30)	Componente B	9
XMEAS(31)	Componente C	9
XMEAS(32)	Componente D	9
XMEAS(33)	Componente E	9
XMEAS(34)	Componente F	9
XMEAS(35)	Componente G	9
XMEAS(36)	Componente H	9
XMEAS(37)	Componente D	11
XMEAS(38)	Componente E	11
XMEAS(39)	Componente F	11
XMEAS(40)	Componente G	11
XMEAS(41)	Componente H	11

Fonte: SORAYA et al., 2019.

Além disso, os dados contêm 20 categorias de falhas predeterminadas (D), sendo 5.250.000 dados de treino e 10.080.000 dados de teste, em um total de 5,34 GB de dados, o que é mais que o suficiente para reproduzir um contexto de Big Data (SOARES, 2017). O conjunto de dados foi disponibilizado para uso livre pelo *Office of Naval Research, Human Bioengineered Systems* (ONR 341).

Tabela 4 – Tipos de Falhas Predeterminadas.

Falha	Interpretação	Tipo de perturbação
D(1)	Razão de alimentação A/C, composição de B constante	Degrau
D(2)	Composição de B, razão de alimentação A/C constante	Degrau
D(3)	Temperatura de alimentação de D	Degrau
D(4)	Temperatura da água de alimentação do reator	Degrau
D(5)	Temperatura da água de alimentação do condensador	Degrau
D(6)	Perda de alimentação de A	Degrau
D(7)	Perda de pressão do cabeçalho C — Disponibilidade reduzida	Degrau
D(8)	Composição de alimentação de A, B, C	Variação Aleatória
D(9)	Temperatura de alimentação de D	Variação Aleatória
D(10)	Temperatura da alimentação de C	Variação Aleatória
D(11)	Temperatura da entrada da água de resfriamento do reator	Variação Aleatória
D(12)	Temperatura da entrada da água de resfriamento do condensador	Variação Aleatória
D(13)	Cinética de Reação	Desvio Lento
D(14)	Válvula de água de resfriamento do reator	Emperrando
D(15)	Válvula de água de resfriamento do condensador	Emperrando
D(16)	Não Determinado	Não Determinado
D(17)	Não Determinado	Não Determinado
D(18)	Não Determinado	Não Determinado
D(19)	Não Determinado	Não Determinado
D(20)	Não Determinado	Não Determinado

Fonte: YUAN; TIAN, 2019

3.5 PYTHON E SUAS RESPECTIVAS BIBLIOTECAS

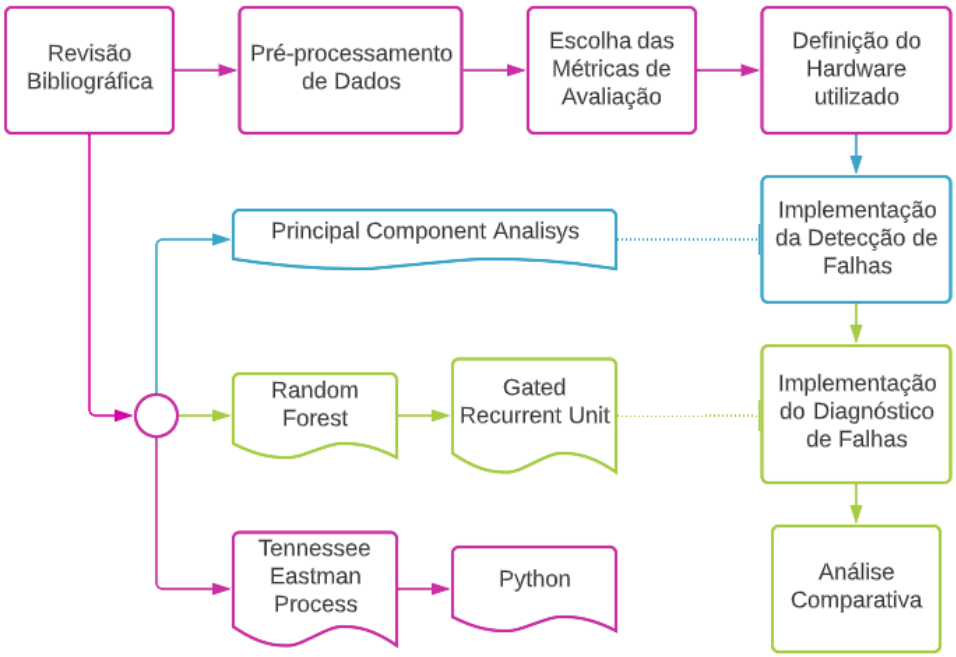
Por conta da linguagem de programação *Python* se tornar cada vez mais popular, principalmente na área da engenharia, e ser gratuita, aliado a um desenvolvimento crescente de novas bibliotecas, graças a uma ampla comunidade, esse trabalho busca implementar diversas bibliotecas em *Python* para a construção do modelo de FDD para o TEP.

Algumas bibliotecas que apresentam potencial para realização deste trabalho incluem a *NumPy*, para computação científica e auxiliar no desenvolvimento do PCA com cálculos de álgebra linear, *matplotlib*, para construção dos gráficos e diagramas, *pyreadr* e *Pandas*, para leitura e manipulação de dados, *Scikit-learn*, para aplicação da RF e das métricas de avaliação, e *Keras* para aplicação da GRU.

4 METODOLOGIA

As etapas para realização deste trabalho podem ser resumidas conforme a Figura 5, primeiramente ocorreu a revisão bibliográfica envolvendo estratégias de FDD e as técnicas de PCA focado na detecção de falhas e no GRU e RF focados no diagnóstico de falhas. Em seguida, foi construída a rotina computacional do PCA para etapa de detecção e seguidamente, as rotinas computacionais da RF e do GRU para a etapa de diagnóstico de falhas e, logo após uma análise comparativa entre às duas técnicas de diagnóstico.

Figura 5 – Metodologia desenvolvida no trabalho.



Fonte: Autor, 2021.

4.1 PRÉ-PROCESSAMENTO DE DADOS

Uma das etapas mais importantes em qualquer atividade que envolva ciência de dados é o tratamento dos dados, como a normalização, tratamento de valores ausentes, diminuição da dimensionalidade, eliminação de valores atípicos, etc. (ALIABADI, 2020). Os dados simulados já vem tratados, então no escopo deste trabalho foi escolhido aplicar apenas a normalização removendo a média e dimensionando os elementos para terem variância unitária, através da transformação: (BUITINCK et al., 2013)

$$z_i = \frac{x_i - \mu_i}{\sigma_i} \tag{12}$$

4.2 ESCOLHA DAS MÉTRICAS DE AVALIAÇÃO

4.2.1 Avaliação da detecção de falhas

Para avaliar a eficácia do PCA, além da análise gráfica das cartas de controle que demonstram a capacidade da técnica, é possível aplicar a técnica aos dados sem falhas, o que permite analisar a taxa de alarmes falsos, e para cada falha é utilizado a taxa de dados acima dos limites T^2 e Q.

4.2.2 Avaliação do diagnóstico de falhas

Já na etapa de diagnóstico de falhas, a eficiência dos modelos foi determinada utilizando as métricas de classificação a seguir:

- Acurácia: A acurácia é a proporção de previsões corretas do classificador e seja TP as amostras positivas classificados corretamente, TN as amostras falsas classificados corretamente, FP os falsos positivos e FN os falsos negativos, calcula-se a acurácia através da fórmula: (FAWCETT, 2006)

$$acuracia = \frac{TP + TN}{TP + TN + FP + FN} \tag{13}$$

- Precisão: A precisão é a habilidade do classificador de não classificar como falsa uma amostra verdadeira, ela é calculada da seguinte maneira: (FAWCETT, 2006)

$$precisao = \frac{TP}{TP + FP} \tag{14}$$

- *Recall*: é a habilidade do classificador de achar todas as amostras positivas, seja FN o número de falsos negativos, ela é dada por: (FAWCETT, 2006)

$$recall = \frac{TP}{TP + FN} \tag{15}$$

- *F1-score*: É a média harmônica ponderada entre a precisão e o *recall* (FAWCETT, 2006);

- Área Sob a Curva ROC (*Receiver Operating Characteristic)*: A curva ROC é uma função bidimensional que consegue representar o desempenho de um classificador, mas para comparar classificadores é necessário um valor escalar e uma maneira comum de se conseguir esse valor é calculando a Área Sob a Curva ROC (AUC), sendo um número entre 0 e 1, quanto maior sua área, maior a performance em média do classificador (FAWCETT, 2006).

4.3 DEFINIÇÃO DO *HARDWARE* UTILIZADO

A detecção de falhas com PCA e o diagnóstico de falhas por RF foram desenvolvidos em um computador com processador *AMD Ryzen 3 3 100 4-Core* SSD de 256 GB e 16 GB de memória RAM. Já o diagnóstico de falhas por GRU, por ser uma técnica de aprendizado profundo, lida melhor com um número elevado de dados, mas necessita de uma maior performance computacional, por isso foi utilizado um computador com processador *Intel Core* i5, 8.ª geração, com SSD de 256 GB e 32 GB de memória RAM.

5 RESULTADOS E DISCUSSÃO

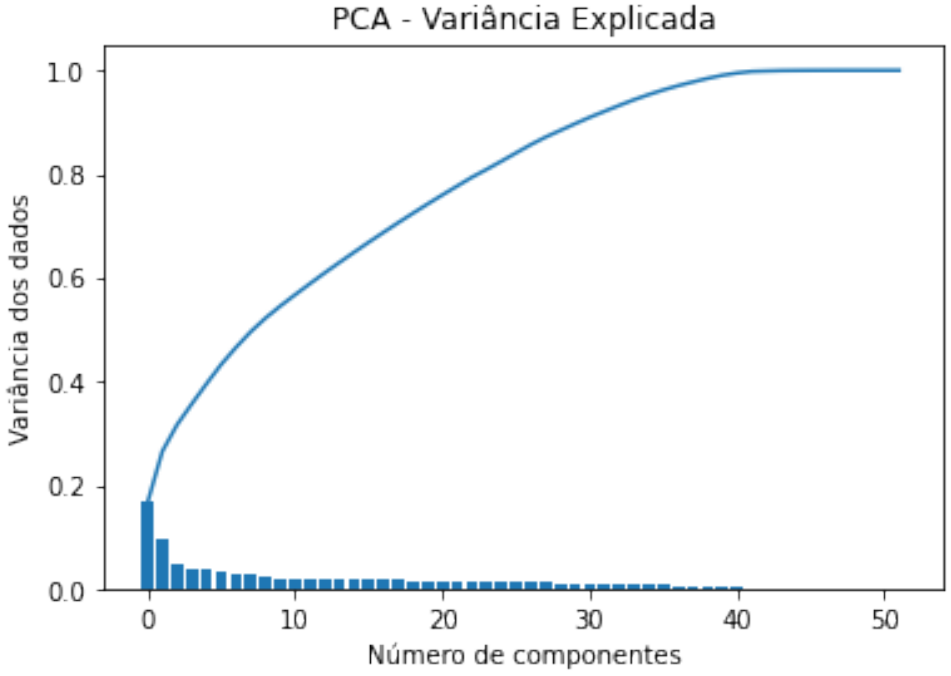
Os resultados serão mostrados inicialmente pela análise da detecção de falhas por PCA aplicado a cada uma das 20 falhas, os estados sistemáticos que se afastam do normal, enumerados na Tabela 4. Logo após serão analisados os diagnósticos das falhas por RF e GRU para demostrar a capacidade de aprendizado de máquina e aprendizado profundo, que serão comparados.

5.1 DETECÇÃO DE FALHAS

O PCA é uma transformação do espaço vetorial usado para transformar o espaço multivariado em um subespaço que tenta preservar o máximo da variância do espaço original no menor número de dimensões, então foi primeiramente definido que o número dos componentes do PCA foi decidido como a quantidade que explica 95% da variância dos dados que segundo a Figura 6, acontece com 40 componentes, diminuindo a dimensionalidade de um total de 52 variáveis (GARCIA-ALVAREZ, 2009).

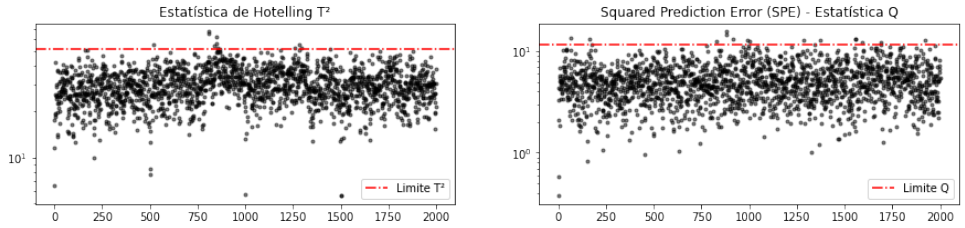
O próximo passo na análise do PCA para detecção de falhas é descobrir a capacidade da técnica de não apontar falsos positivos ao classificar como falha um dado normal de processo, o que pode ser chamado taxa de alarmes falsos, isso é feito aplicando o PCA nos dados sem falhas, como é feito na Figura 7.

Figura 6 – Variância explicada pelo PCA.



Fonte: Autor, 2022.

Figura 7 – PCA aplicado aos dados sem falhas.



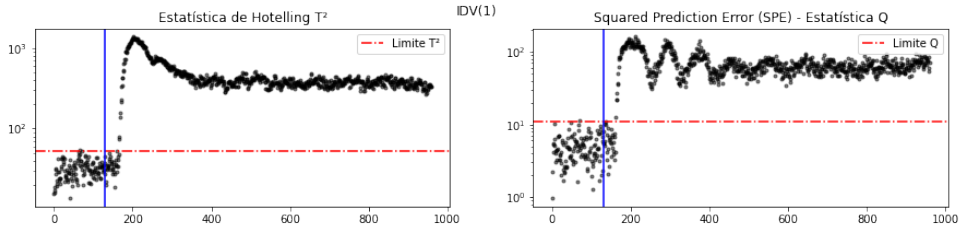
Fonte: Autor, 2022.

Os resultados indicaram que a Estatística de *Hotelling* apontou como falha 0.851% dos dados e a Estatística Q 2.7% dos dados, demonstrando uma taxa pequena de alarmes falsos, e, portanto uma boa taxa de confiabilidade.

Após aferir a taxa de alarmes falsos, O PCA foi então aplicado para todas as falhas individualmente, para analisar a aplicabilidade de detectar cada falha, utilizando o T^2 e Q, que aponta como falha todo valor acima dos limites e, além disso, foi feita uma análise da contribuição de cada variável do processo na falha, utilizando o *Squared Prediction Error* que representa o quanto cada variável foi responsável pelo estado fora do normal.

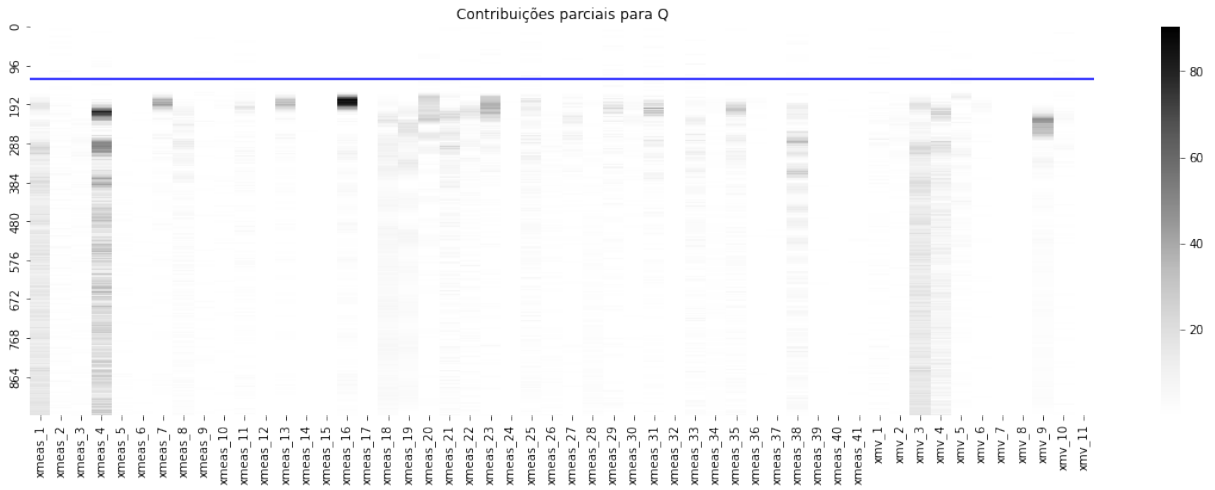
As Figuras 8 e 9 demonstram o funcionamento do modelo, classificando mais de 99% dos dados acima da linha limite, vale denotar que na simulação dos dados, os primeiros 130 dados ainda constituíam valores de variáveis considerados normais de processo e a aplicação da técnica só é valida após esses dados, demarcados pela linha azul.

Figura 8 – PCA aplicado à falha 1.



Fonte: Autor, 2022.

Figura 9 – Contribuições parciais Q na falha 1.



Fonte: Autor, 2022.

Os resultados estão compilados na Tabela 5.

Tabela 5 – Resultado da detecção de falhas por PCA.

Falha	T^2	Q
D(1)	0.99250	0.99500
D(2)	0.98625	0.97875
D(3)	0.01250	0.01750
D(4)	0.35375	1.00000
D(5)	0.26500	0.20625
D(6)	0.99125	1.00000
D(7)	1.00000	0.96750
D(8)	0.97500	0.94000
D(9)	0.01250	0.01375
D(10)	0.06625	0.28000
D(11)	0.41375	0.65750
D(12)	0.97875	0.96250
D(13)	0.91875	0.92625
D(14)	0.99500	0.95250
D(15)	0.01375	0.01250
D(16)	0.05000	0.36125
D(17)	0.70250	0.89625
D(18)	0.92000	0.93375
D(19)	0.10250	0.13250
D(20)	0.20625	0.57000

Fonte: Autor, 2022.

As falhas que foram bem detectadas apresentam resultados satisfatórios, apontando como

falha mais de 90% dos dados, mas como cada falha tem suas peculiaridades, nem todas foram bem detectadas, mais especificamente as apresentadas na Tabela 6.

Tabela 6 – Falhas que não foram bem detectadas pelo PCA.

Falha	T^2	Q
D(3)	0.01250	0.01750
D(4)	0.35375	1.00000
D(5)	0.26500	0.20625
D(9)	0.01250	0.01375
D(10)	0.06625	0.28000
D(11)	0.41375	0.65750
D(15)	0.01375	0.01250
D(16)	0.05000	0.36125
D(19)	0.10250	0.13250
D(20)	0.20625	0.57000

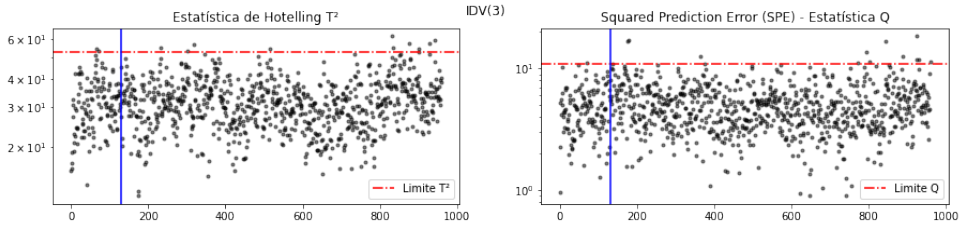
Fonte: Autor, 2022.

As falhas que não foram bem detectadas foram:

- A falha D(3) (Figura 10) referente à Temperatura de alimentação de D na corrente 2 apresentou pertubação do tipo degrau;
- A falha D(4) (Figura 11) referente à Temperatura de água de resfriamento do reator apresentou pertubação do tipo degrau, mas só apresentou problemas no T^2, já que o Q exibiu alta precisão, possivelmente porque as alterações ocorreram no espaço do PCA correspondente aos valores singulares menores;
- A falha D(5) (Figura 12) referente à Temperatura da água de resfriamento do condensador apresentou pertubação do tipo degrau, e mostrou uma detecção satisfatória inicialmente, mas possivelmente na simulação desta falha eventualmente os dados retornaram ao estado estacionário, não sendo mais vistos como falha pelo PCA;
- A falha D(9) (Figura 13) referente à Temperatura de alimentação de D na corrente 2 apresentou pertubação aleatória;
- A falha D(10) (Figura 14) referente à Temperatura de alimentação de C na corrente 5 apresentou pertubação aleatória;
- A falha D(11) (Figura 15) referente à Temperatura da água de resfriamento do condensador apresentou pertubação aleatória;

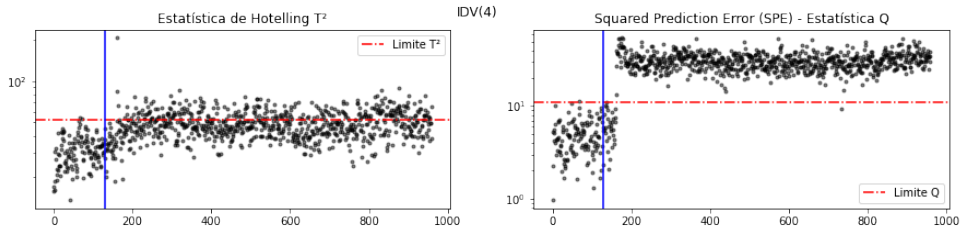
- A falha D(15) (Figura 16) referente à Válvula de água para resfriamento do condensador emperrando;
- As falhas D(16) (Figura 17) e D(19) (Figura 18) em que o motivo da falha e, da baixa detecção são inconclusivos;
- A falha D(20) (Figura 19) apresenta uma natureza oscilatória na percepção da falha e tem motivo indeterminado, mas através das contribuições parciais (Figura 20), é possível notar que a maior contribuição foi por parte do XMEAS(20) sendo o trabalho do compressor e XMV(5) sendo a válvula de reciclo do compressor, então mesmo com a capacidade de detecção baixa, o PCA permitiu inferir que a falha foi simulada com problemas ou no compressor, ou em seus periféricos;

Figura 10 – PCA aplicado à falha 3.



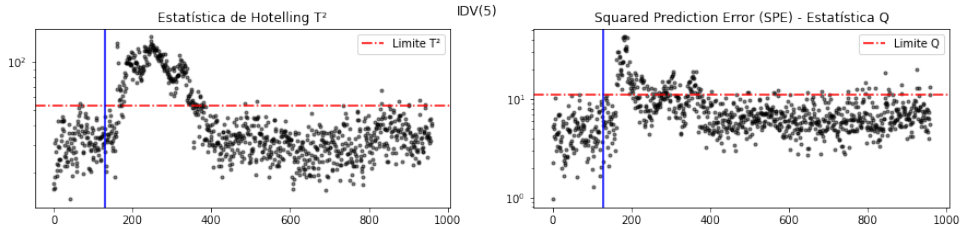
Fonte: Autor, 2022.

Figura 11 – PCA aplicado à falha 4.



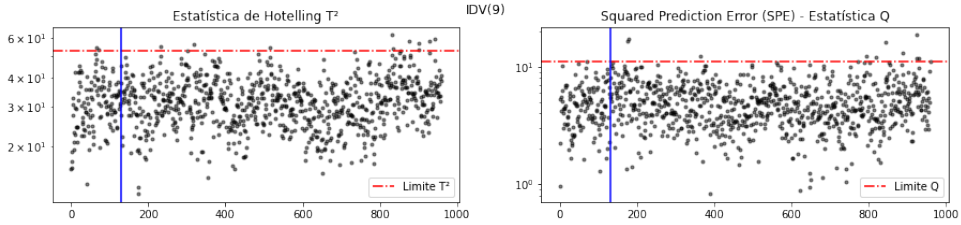
Fonte: Autor, 2022.

Figura 12 – PCA aplicado à falha 5.



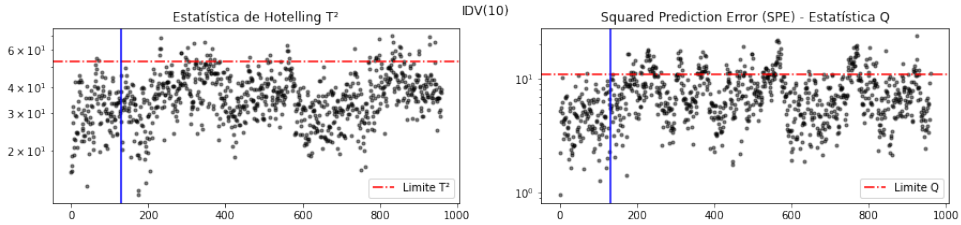
Fonte: Autor, 2022.

Figura 13 – PCA aplicado à falha 9.



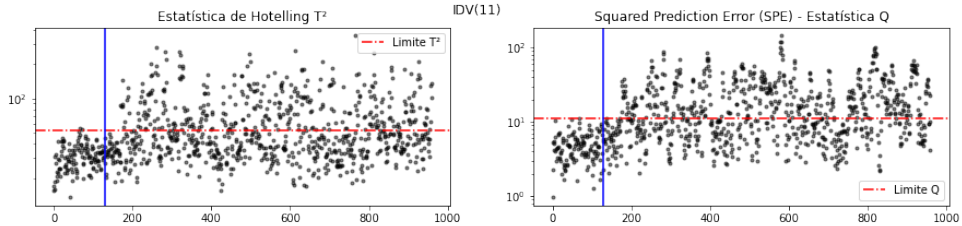
Fonte: Autor, 2022.

Figura 14 – PCA aplicado à falha 10.



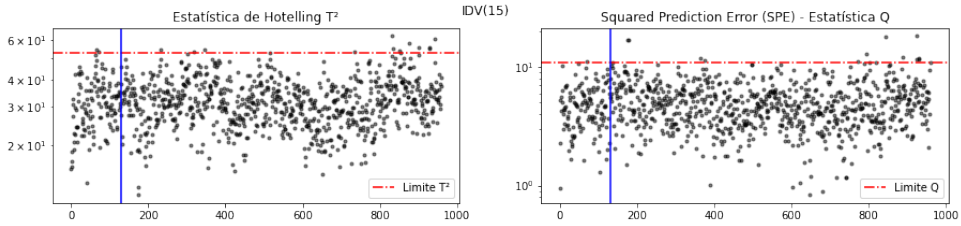
Fonte: Autor, 2022.

Figura 15 – PCA aplicado à falha 11.



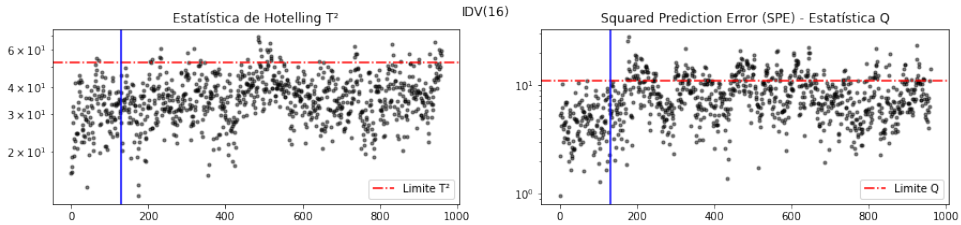
Fonte: Autor, 2022.

Figura 16 – PCA aplicado à falha 15.



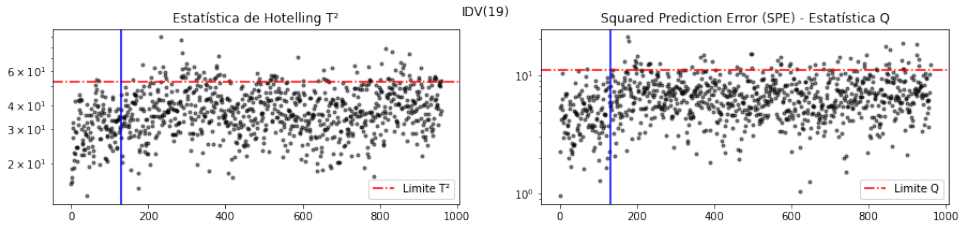
Fonte: Autor, 2022.

Figura 17 – PCA aplicado à falha 16.



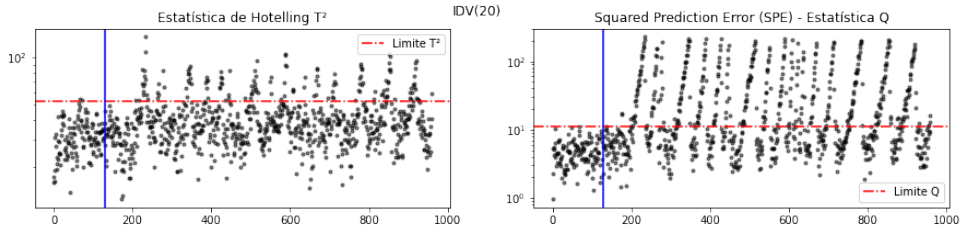
Fonte: Autor, 2022.

Figura 18 – PCA aplicado à falha 19.



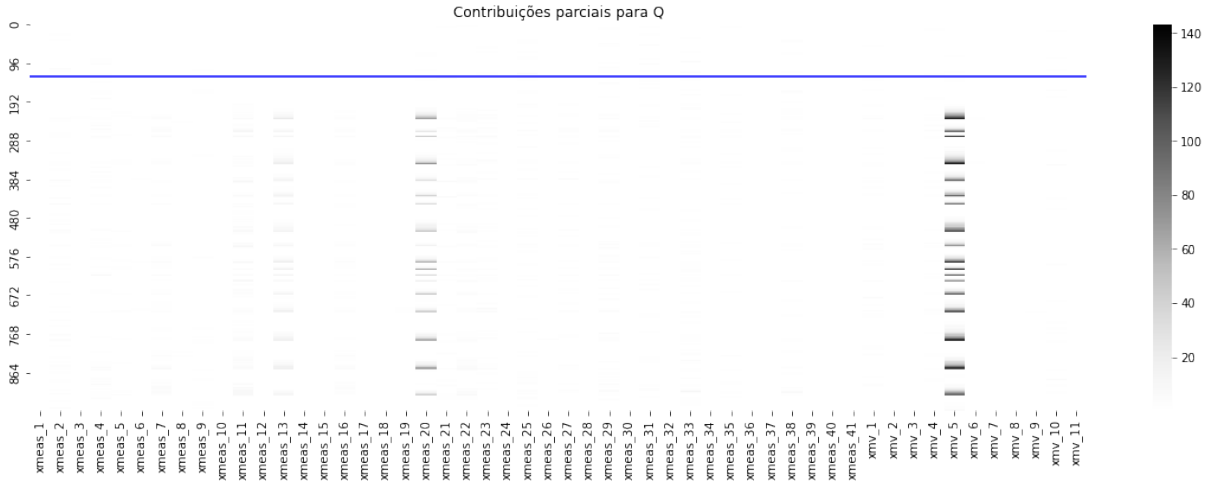
Fonte: Autor, 2022.

Figura 19 – PCA aplicado à falha 20.



Fonte: Autor, 2022.

Figura 20 – Contribuições parciais Q na falha 20.



Fonte: Autor, 2022.

As demais figuras que demonstram os resultados para as falhas não discutidas se encontram no Apêndice A.

5.2 DIAGNÓSTICO DE FALHAS

5.2.1 *Random Forest*

Primeiramente o modelo foi treinado com todas as falhas, para avaliar a capacidade média de classificação, e então apenas com as falhas que o modelo pode diagnosticar para um aumento da clareza dos resultados.

Os resultados para a aplicação da RF utilizando *Scikit-learn* em todas as falhas se encontram na Tabela 7, o único parâmetro de arquitetura do modelo que apresentou mudança significativa nos resultados foi a profundidade máxima da árvore (*max depth*) que foi mantida em 2, ou seja, a árvore fará duas perguntas antes de tomar a decisão de qual classificação será aplicada a determinados dados, quanto maior a profundidade, mais complexa é a árvore tem uma maior capacidade de chegar na resposta, porém quanto maior a profundidade, mais fácil é de o modelo viciar nos dados de treino e ser incapaz de predizer os de treino, no chamado *overfitting*.

Tabela 7 – Resultados do diagnóstico por RF.

Falha	Precisão	*Recall*	*F1-score*
1	0.91	1.00	0.95
2	0.98	0.80	0.88
3	0.83	1.00	0.91
4	0.57	0.77	0.66
5	0.00	0.00	0.00
6	0.96	0.83	0.89
7	0.96	0.81	0.88
8	0.74	0.00	0.01
9	0.15	1.00	0.26
10	0.00	0.00	0.00
11	0.00	0.00	0.00
12	0.42	0.19	0.26
13	0.58	0.36	0.44
14	0.00	0.00	0.00
15	0.50	1.00	0.67
16	0.00	0.00	0.00
17	1.00	1.00	1.00
18	0.73	0.63	0.68
19	0.75	0.99	0.85
20	0.99	1.00	1.00
Média	0.55	0.57	0.52
Acurácia			0.57
AUC			0.77

Fonte: Autor, 2022.

Com uma acurácia de 57%, o modelo não parece promissor, mas a etapa de detecção mostrou que as falhas divergem bastante em seus comportamentos, então foram retiradas as classes com baixa precisão, assinaladas na Tabela 8, algumas das quais também apresentaram baixo resultado na detecção, como a 5, 10, 11 e 16, e o resultado exibido na Tabela 9 foi bastante positivo, mostrando que o modelo de aprendizado das máquinas por RF foi eficiente para 70% das classes, ressaltando que a rotina computacional demorou 13 minutos para completar com todas as falhas e 8 ao retirar as com resultado baixo.

Tabela 8 – Falhas que não foram bem diagnosticados pela RF.

Falha	Precisão	*Recall*	*F1-score*
5	0.00	0.00	0.00
8	0.74	0.00	0.01
10	0.00	0.00	0.00
11	0.00	0.00	0.00
14	0.00	0.00	0.00
16	0.00	0.00	0.00

Fonte: Autor, 2022.

Tabela 9 – Resultados do diagnóstico por RF sem classes com resultados ruins.

Falha	Precisão	*Recall*	*F1-score*
1	0.96	1.00	0.98
2	0.86	0.81	0.83
3	0.65	1.00	0.79
4	0.98	1.00	0.99
6	1.00	0.83	0.91
7	1.00	0.82	0.90
9	0.69	1.00	0.82
12	0.83	0.27	0.41
13	0.84	0.53	0.65
15	0.78	1.00	0.87
17	1.00	1.00	1.00
18	0.81	0.64	0.71
19	0.76	1.00	0.86
20	1.00	1.00	1.00
Média	0.87	0.85	0.84
Acurácia			0.85
AUC			0.92

Fonte: Autor, 2022.

Vale salientar que a falha D(12) não apresentou bons resultados, mas retirá-la diminuiu a eficiência geral do modelo.

5.2.2 *Gated Recurrent Unit*

Os dados pré-processados foram utilizados na GRU para o diagnóstico das falhas utilizando a arquitetura definida no trabalho de Yuan e Tian (2019), que comparou diversas arquiteturas de GRU para o TEP, mudando a quantidade de épocas de treinamento para 50 épocas que duraram 200 segundos cada para treinar, em um total de 2h e 47 minutos.

Os hiperparâmetros avaliados foram o inicializador, a função de ativação e o algoritmo de otimização, tendo sido determinado de maneira heurística que a inicialização dos pesos das redes foi realizada pelo *Xavier normal initializer* (GLOROT; BENGIO, 2010), usado para auxiliar a convergência do treino, a função de ativação foi a *softmax* (BRIDLE, 1989), sendo a função em execução nos neurônios, e o algoritmo de otimização utilizado foi o ADAM (KINGMA; BA, 2015), cujo objetivo é diminuir o erro entre os resultados obtidos por uma rede em comparação com os resultados desejados.

Os resultados demonstrados na Tabela 10 foram bastante satisfatórios, não apresentando resultado abaixo de 81% em nenhuma das falhas, nem mesmo as que apresentaram dificuldade para as outras técnicas já bem estabelecidas.

Tabela 10 – Resultados do diagnóstico por GRU.

Falha	Precisão	*Recall*	*F1-score*
1	1.00	1.00	1.00
2	1.00	1.00	1.00
3	1.00	1.00	1.00
4	0.98	1.00	0.99
5	1.00	0.98	0.99
6	1.00	1.00	1.00
7	1.00	1.00	1.00
8	1.00	1.00	1.00
9	1.00	0.77	0.87
10	0.81	1.00	0.90
11	1.00	1.00	1.00
12	1.00	1.00	1.00
13	1.00	0.99	0.99
14	1.00	1.00	1.00
15	0.83	1.00	0.90
16	1.00	0.79	0.88
17	1.00	1.00	1.00
18	0.99	1.00	1.00
19	1.00	1.00	1.00
20	1.00	1.00	1.00
Média	0.986	0.982	0.981
Acurácia			0.982
AUC			0.988

Fonte: Autor, 2022.

5.2.3 Comparação entre as técnicas implementadas

As métricas mais importantes para a comparação entre a RF e a GRU são a acurácia e a AUC considerando a diferença de performance de *hardware* necessária e o tempo de execução, uma síntese dos resultados se encontra na Tabela 11, é possível ver que o desempenho da GRU foi muito melhor que da RF, mesmo ao retirar as falhas, mas é preciso considerar que para isso foi necessário o dobro de memória RAM e mais de 10 vezes o tempo de execução.

Tabela 11 – Comparação das técnicas de diagnóstico de falha.

Técnica	Acurácia	AUC	Tempo (min)	Hardware
RF (20 Falhas)	0.57	0.77	13	AMD Ryzen 3 3100, 16 GB de RAM
RF (14 Falhas)	0.84	0.92	8	AMD Ryzen 3 3100, 16 GB de RAM
GRU	0.9818	0.9904	167	Intel Core i5, 8.ª gen, 32 GB de RAM

Fonte: Autor, 2022.

6 CONCLUSÃO

No presente trabalho, foram estudadas as técnicas do *Principal Component Analisys* um procedimento matemático, da *Random Forest* uma técnica de aprendizado de máquina e da *Gated Recurrent Unit* como parte do aprendizado profundo. As técnicas foram utilizadas para a detecção e diagnóstico das falhas no processo químico conhecido como *Tennessee Eastman Process* escolhido por ser bastante completo e fornecer quantidade de dados suficiente para o contexto do *Big Data*.

A aplicação do PCA permitiu entender que o processo poderia ser traduzido por 40 componentes principais em vez das 52 variáveis do processo e foi possível detectar com propriedade 11 das 20 falhas, pelo comportamento complexo de algumas das falhas e pela técnica ser umas das mais simples das verdadeiras análises multivariadas.

Já para o diagnóstico das falhas que seriam detectadas pelo PCA, a RF apresentou um bom resultado apenas para 14 das 20 falhas, já a GRU se provou como um modelo mais inovador, classificando de maneira bastante satisfatória todas as falhas, mas para isso precisou do dobro de desempenho computacional e mais de 10 vezes o tempo de execução.

Foi avaliado também a aplicabilidade da linguagem de programação *Python* e suas bibliotecas para manipulação dos dados e construção das técnicas. A linguagem mostrou seu potencial para atender as demandas do FDD.

Sendo assim, este trabalho atingiu todos os objetivos propostos, sendo possível aplicar possível consolidar dos conhecimentos adquiridos durante a graduação na análise de processos químicos e os conhecimentos nas áreas de aprendizado de máquina e aprendizado profundo, que não são muito explorados durante a graduação, mas foram inicialmente explorados no LABSIA.

REFERÊNCIAS

ALIABADI, M. M. **Process data analytics using Deep Learning Techniques**. Dissertação (Mestrado) — Curso de Computer Science, Graduate School, Wayne State University, Detroit, 2020.

BRIDLE, J. S. Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters. In: TOURETZKY, D. S. (Ed.). **Advances in Neural Information Processing Systems 2, [NIPS Conference, Denver, Colorado, USA, November 27-30, 1989]**. Morgan Kaufmann, 1989. p. 211–217. Disponível em: <http://papers.nips.cc/paper/195-training-stochastic-model-recognition-algorithms-as-networks-can-lead-to-maximum-mutual-informati

BUITINCK, L.; LOUPPE, G.; BLONDEL, M.; PEDREGOSA, F.; MUELLER, A.; GRISEL, O.; NICULAE, V.; PRETTENHOFER, P.; GRAMFORT, A.; GROBLER, J.; LAYTON, R.; VANDERPLAS, J.; JOLY, A.; HOLT, B.; VAROQUAUX, G. API design for machine learning software: experiences from the scikit-learn project. In: **ECML PKDD Workshop: Languages for Data Mining and Machine Learning**. [S.l.: s.n.], 2013. p. 108–122.

CAVALCANTI, F. M.; KOZONOE, C. E.; PACHECO, K. A.; ALVES, R. M. d. B. Application of artificial neural networks to chemical and process engineering. **Deep Learning Applications**, 2021.

CHIANG, L. H.; RUSSELL, E. L.; BRAATZ, R. D. **Fault detection and diagnosis in Industrial Systems**. [S.l.]: Springer, 2001.

CHO, K.; MERRIENBOER, B. van; GÜLÇEHRE, Ç.; BOUGARES, F.; SCHWENK, H.; BENGIO, Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. **CoRR**, abs/1406.1078, 2014. Disponível em: <http://arxiv.org/abs/1406.1078>.

DRAGOGIAS, I. **Fault diagnosis in Industrial Chemical Processes using Machine Learning**. Tese (Doutorado) — Curso de Data Science, School Of Science & Technology, International Hellenic University, Thessaloniki, 2019.

FAWCETT, T. An introduction to roc analysis. **Pattern Recognition Letters**, v. 27, n. 8, p. 861–874, 2006.

FORTUNA, L.; XIBíLIA, M. G.; RIZZO, A.; GRAZIANI, S. **Soft sensors for monitoring and control of Industrial Processes**. [S.l.]: Springer, 2007.

GARCIA-ALVAREZ, D. Fault detection using principal component analysis (pca) in a wastewater treatment plant (wwtp). 01 2009.

GLOROT, X.; BENGIO, Y. Understanding the difficulty of training deep feedforward neural networks. In: TEH, Y. W.; TITTERINGTON, D. M. (Ed.). **Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010**. JMLR.org, 2010. (JMLR Proceedings, v. 9), p. 249–256. Disponível em: <http://proceedings.mlr.press/v9/glorot10a.html>.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, v. 9, p. 1735–80, 12 1997.

KINGMA, D. P.; BA, J. Adam: A method for stochastic optimization. In: BENGIO, Y.; LECUN, Y. (Ed.). **3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings**. [s.n.], 2015. Disponível em: <http://arxiv.org/abs/1412.6980>.

MARINS, M. A.; BARROS, B. D.; SANTOS, I. H.; BARRIONUEVO, D. C.; VARGAS, R. E.; PREGO, T. de M.; LIMA, A. A. de; CAMPOS, M. L. de; SILVA, E. A. da; NETTO, S. L.; AL. et. Fault detection and classification in oil wells and production/service lines using random forest. **Journal of Petroleum Science and Engineering**, v. 197, p. 107879, 2021.

SHAO, B.; HU, X.; BIAN, G.; ZHAO, Y. A multichannel lstm-cnn method for fault diagnosis of chemical process. **Mathematical Problems in Engineering**, v. 2019, p. 1–14, 2019.

SOARES, F. D. R. **Técnicas de Machine Learning Aplicadas a Inferência e Detecção e Diagnóstico de Falhas de Processos Químicos Industriais em Contexto Big Data**. Dissertação (Mestrado) — Curso de Pós-Graduação em Tecnologia de Processos Químicos e Bioquímicos, Escola de Química, Universidade Federal do Rio de Janeiro, Rio de Janeiro, 2017.

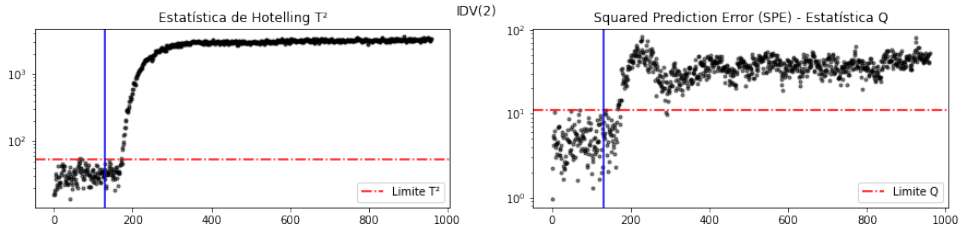
SORAYA, B.; FAOUZI, H. M.; ABDERRAZAK, L. Fault diagnosis of tennessee eastman process based on static pca. **2019 1st International Conference on Sustainable Renewable Energy Systems and Applications (ICSRESA)**, 2019.

YANG, B.-S.; DI, X.; HAN, T. Random forests classifier for machine fault diagnosis. **Journal of Mechanical Science and Technology**, v. 22, n. 9, p. 1716–1725, 2008.

YUAN, J.; TIAN, Y. An intelligent fault diagnosis method using gru neural network towards sequential data in dynamic processes. **Processes**, v. 7, n. 3, p. 152, 2019.

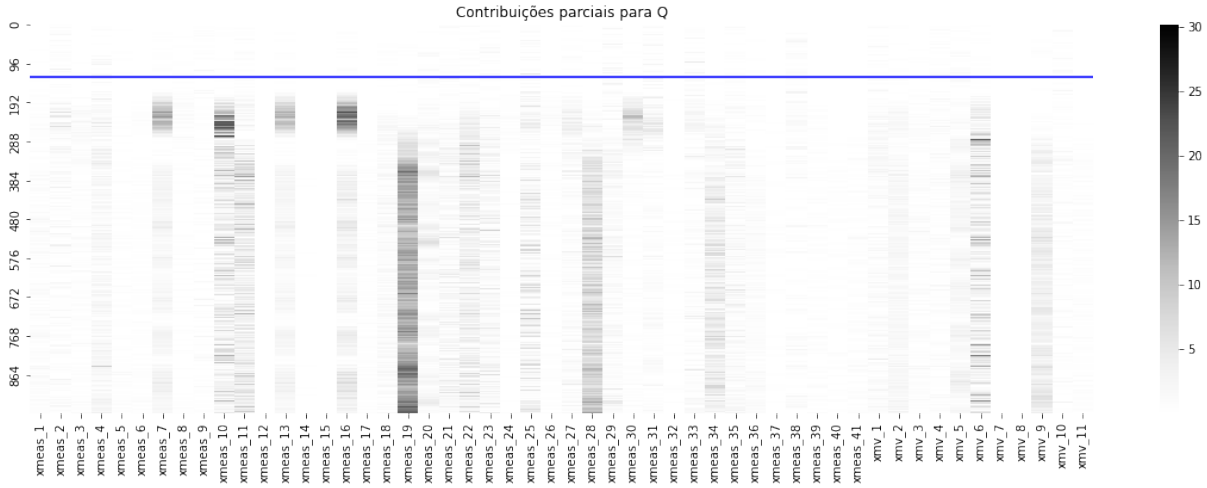
APÊNDICE A – Resultados da detecção de falhas por PCA

Figura 21 – PCA aplicado à falha 2.



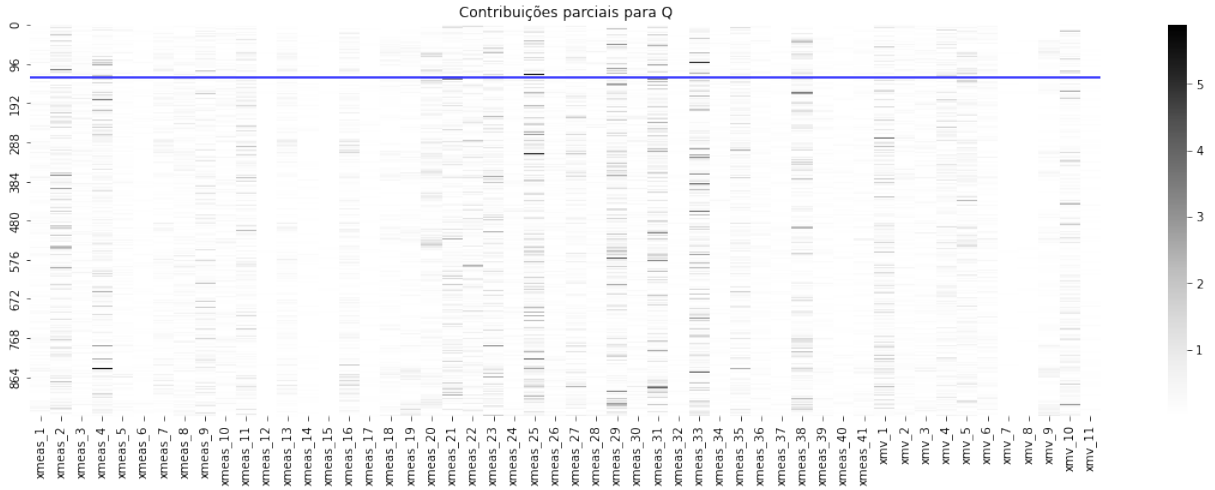
Fonte: Autor, 2022.

Figura 22 – Contribuições parciais Q na falha 2.



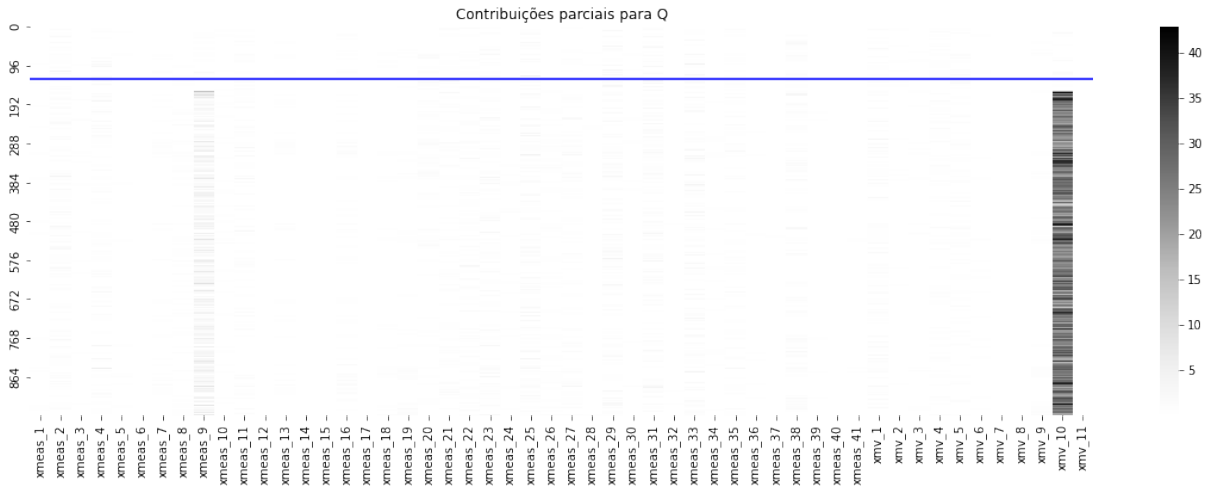
Fonte: Autor, 2022.

Figura 23 – Contribuições parciais Q na falha 3.



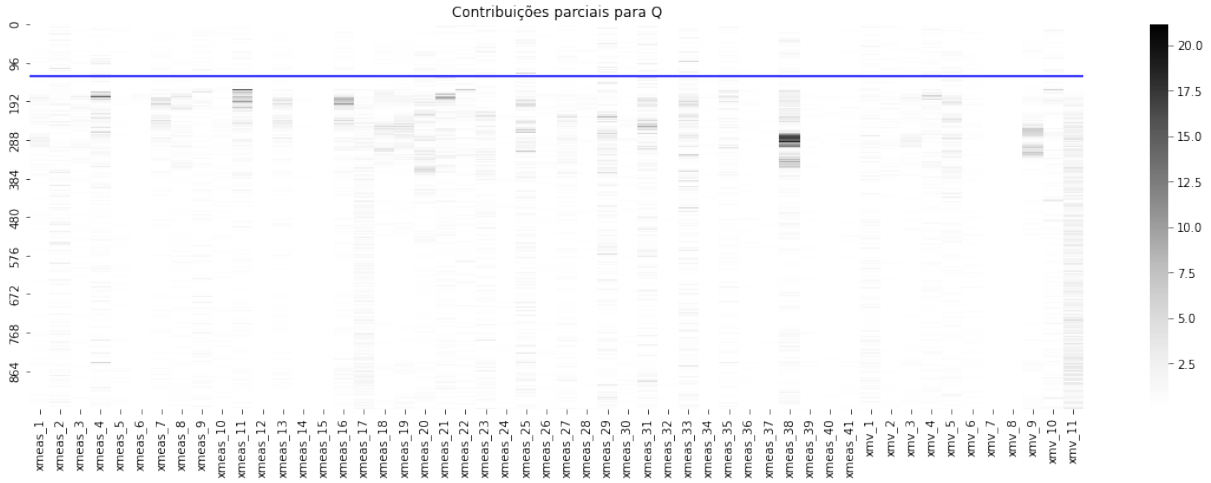
Fonte: Autor, 2022.

Figura 24 – Contribuições parciais Q na falha 4.



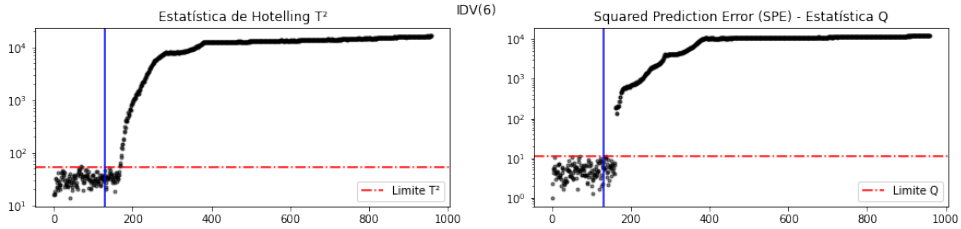
Fonte: Autor, 2022.

Figura 25 – Contribuições parciais Q na falha 5.



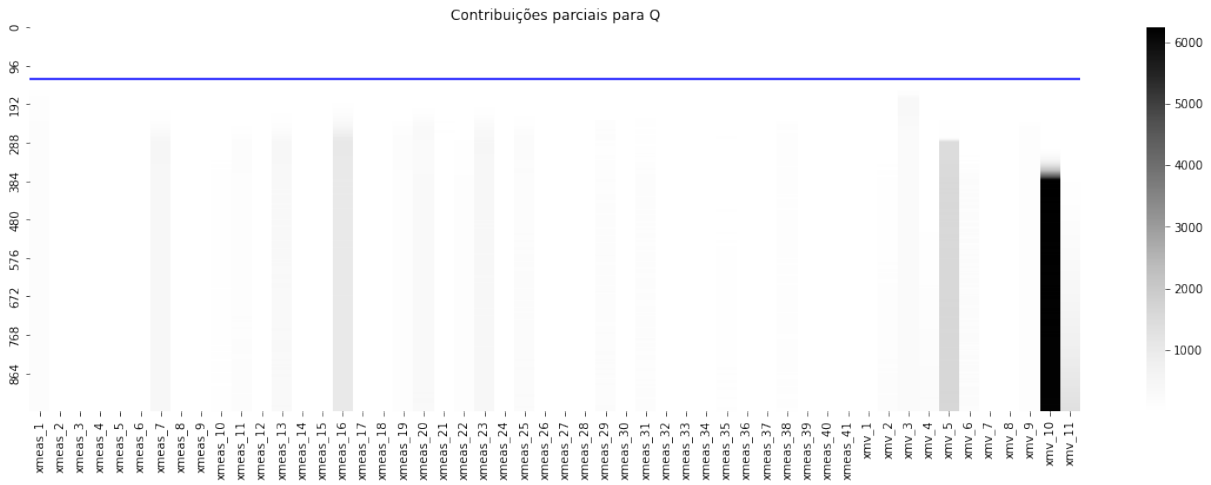
Fonte: Autor, 2022.

Figura 26 – PCA aplicado à falha 6.



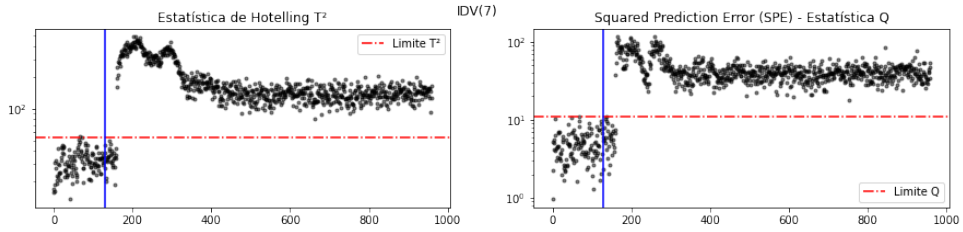
Fonte: Autor, 2022.

Figura 27 – Contribuições parciais Q na falha 6.



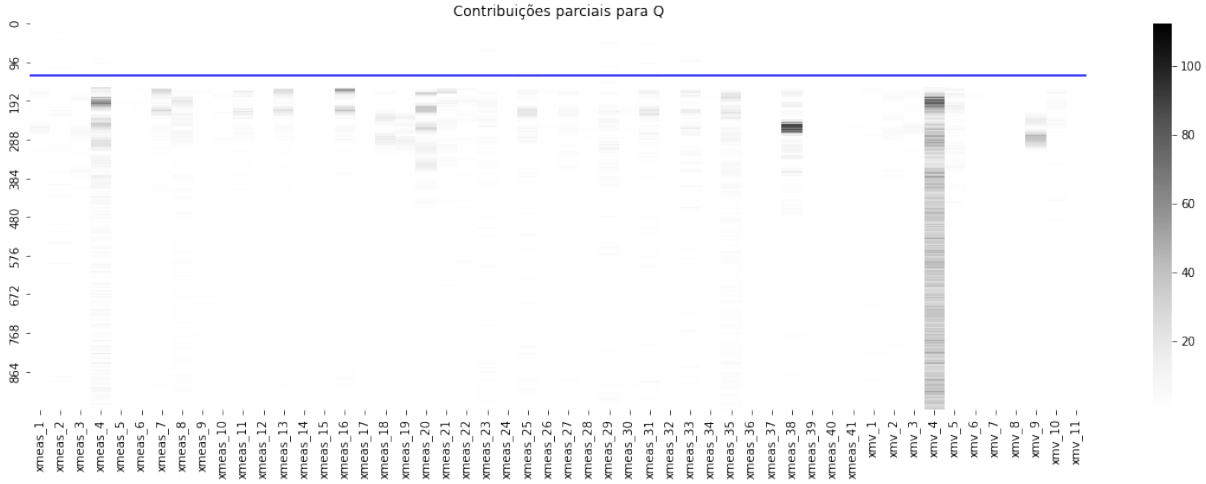
Fonte: Autor, 2022.

Figura 28 – PCA aplicado à falha 7.



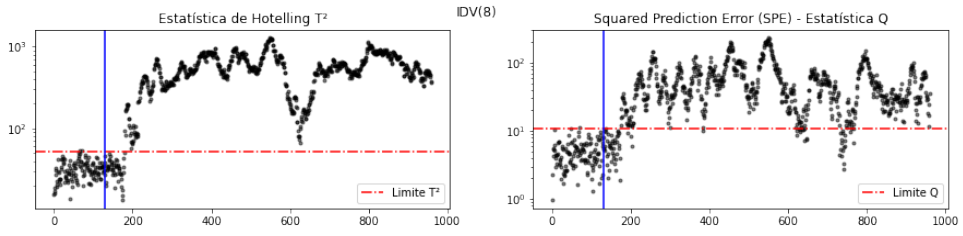
Fonte: Autor, 2022.

Figura 29 – Contribuições parciais Q na falha 7.



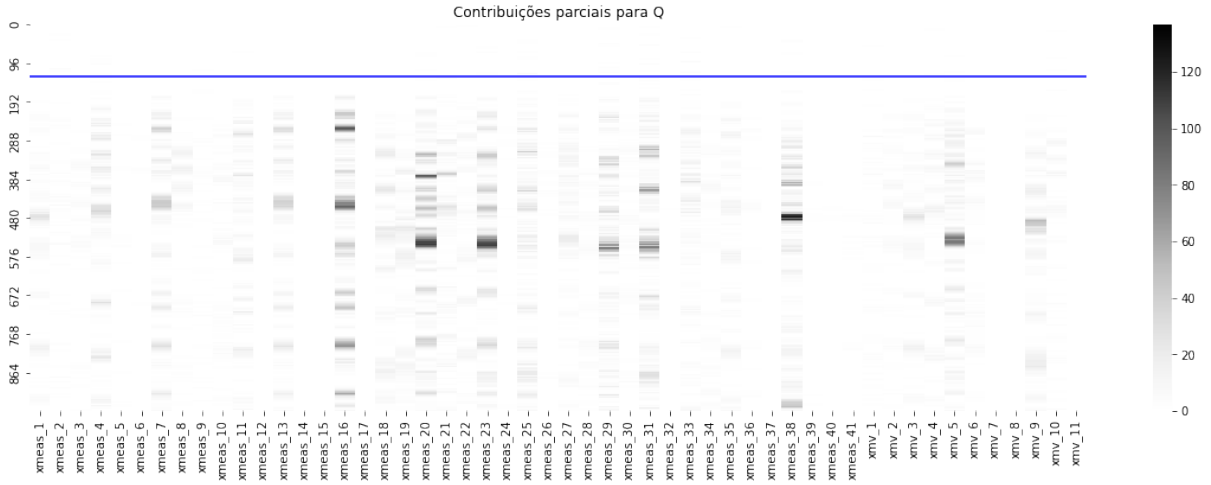
Fonte: Autor, 2022.

Figura 30 – PCA aplicado à falha 8.



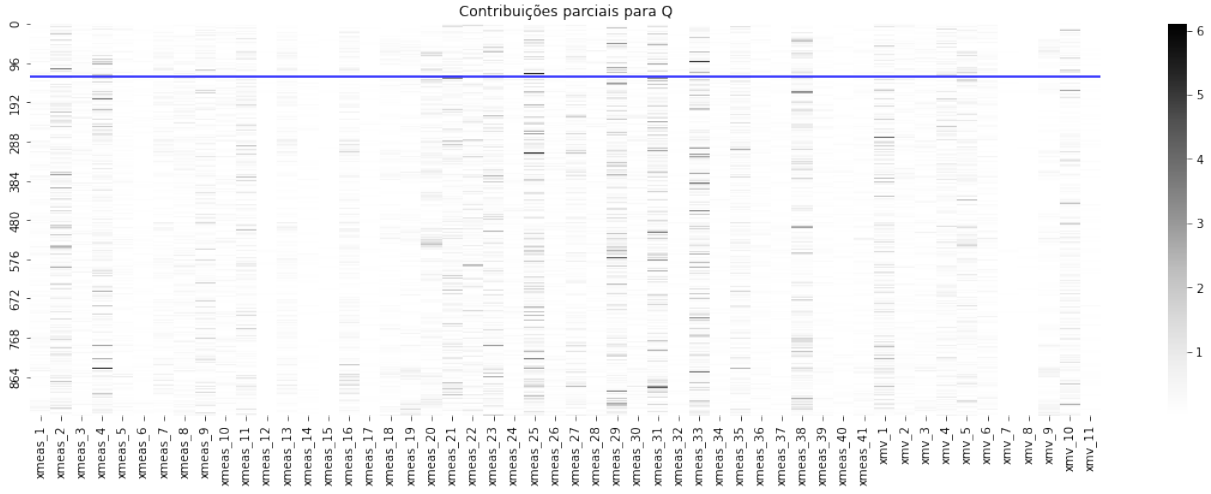
Fonte: Autor, 2022.

Figura 31 – Contribuições parciais Q na falha 8.



Fonte: Autor, 2022.

Figura 32 – Contribuições parciais Q na falha 1.



Fonte: Autor, 2022.

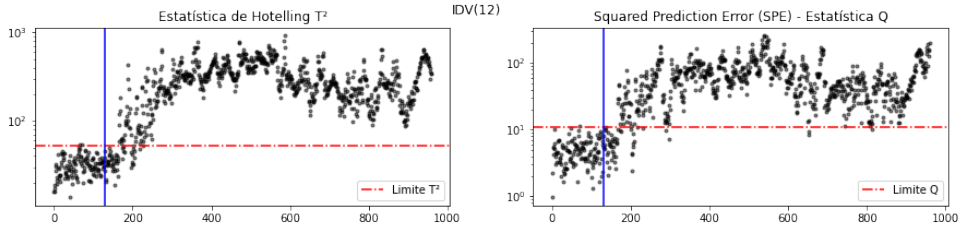
Figura 33 – Contribuições parciais Q na falha 10.

Fonte: Autor, 2022.

Figura 34 – Contribuições parciais Q na falha 11.

Fonte: Autor, 2022.

Figura 35 – PCA aplicado à falha 12.

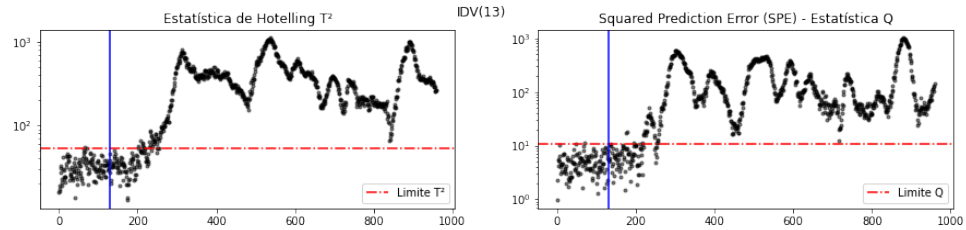


Fonte: Autor, 2022.

Figura 36 – Contribuições parciais Q na falha 12.

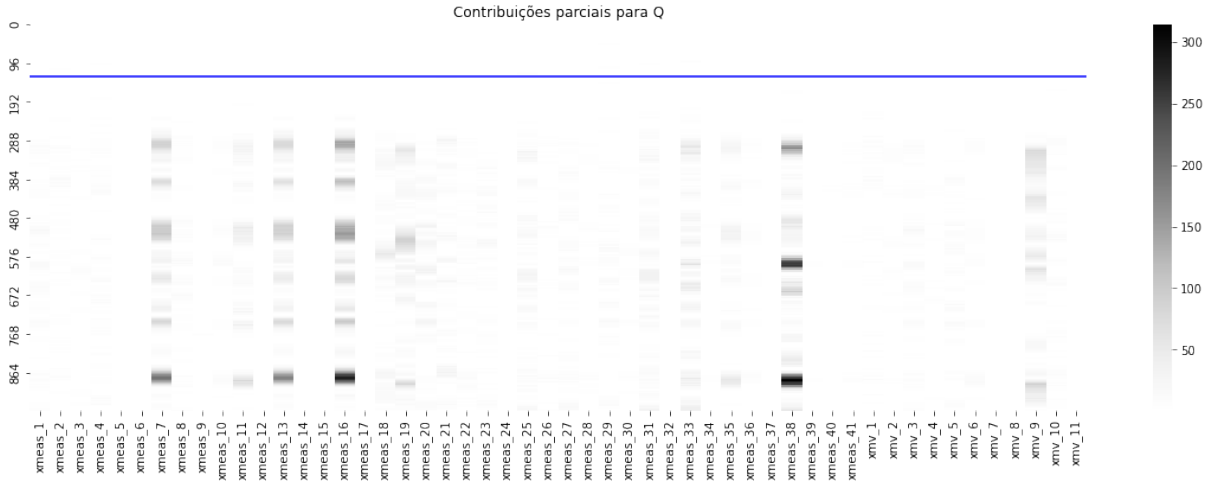
Fonte: Autor, 2022.

Figura 37 – PCA aplicado à falha 13.



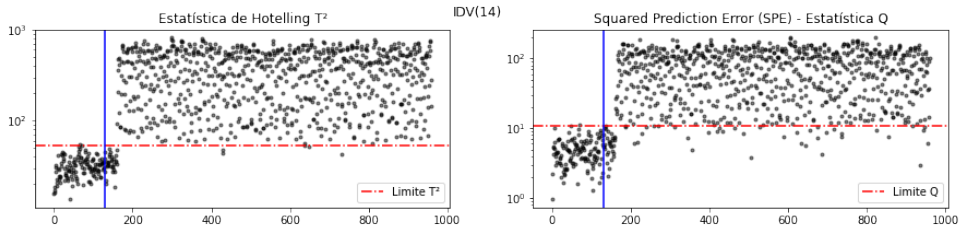
Fonte: Autor, 2022.

Figura 38 – Contribuições parciais Q na falha 13.



Fonte: Autor, 2022.

Figura 39 – PCA aplicado à falha 14.

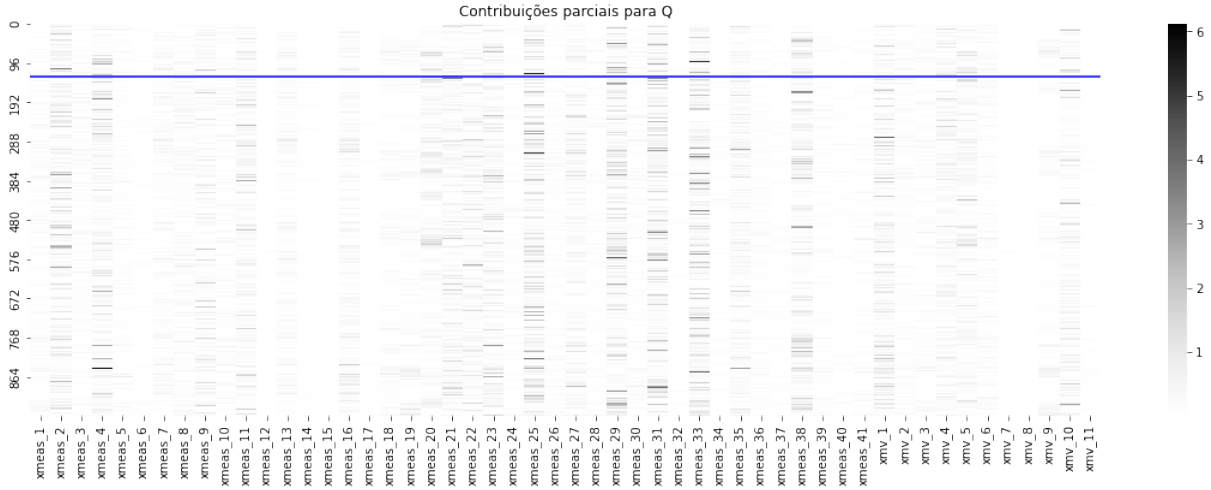


Fonte: Autor, 2022.

Figura 40 – Contribuições parciais Q na falha 14.

Fonte: Autor, 2022.

Figura 41 – Contribuições parciais Q na falha 15.

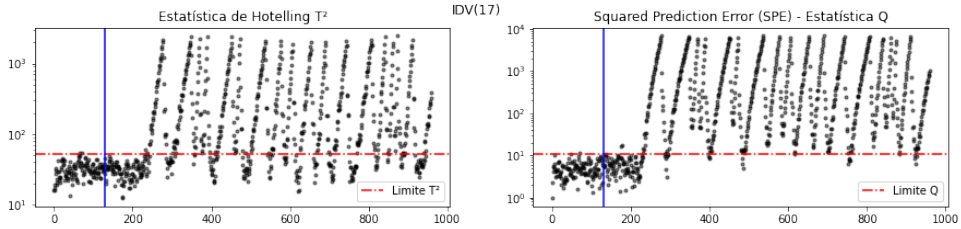


Fonte: Autor, 2022.

Figura 42 – Contribuições parciais Q na falha 16.

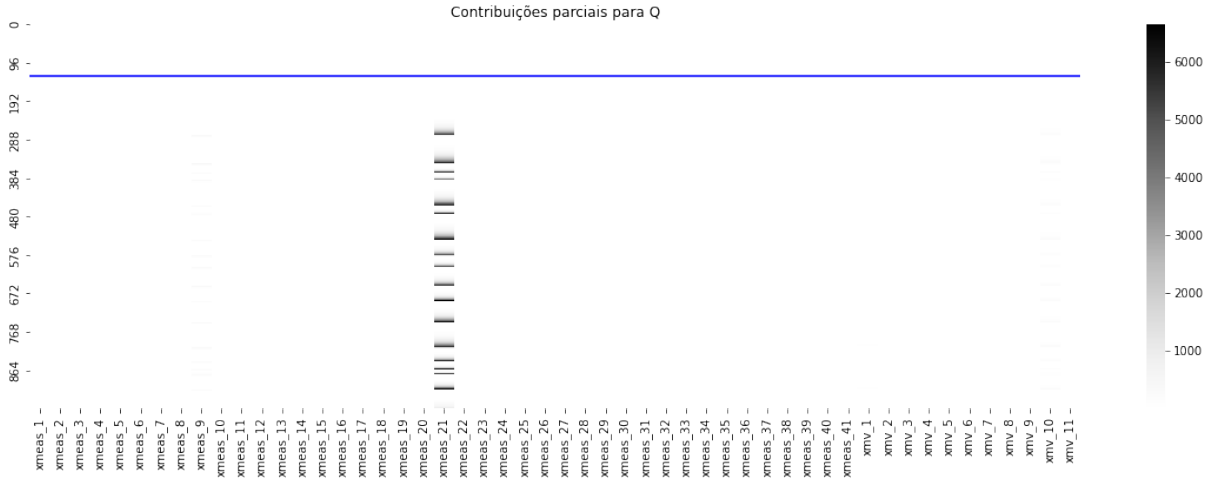
Fonte: Autor, 2022.

Figura 43 – PCA aplicado à falha 17.



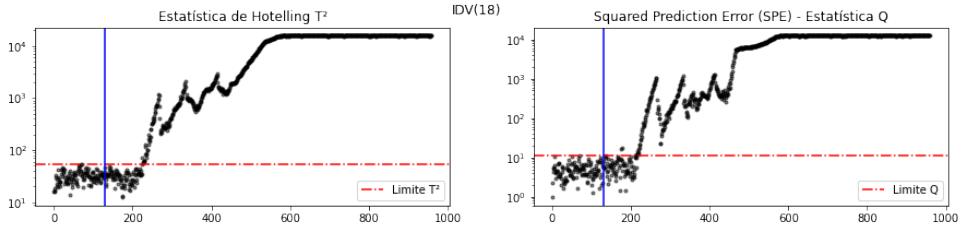
Fonte: Autor, 2022.

Figura 44 – Contribuições parciais Q na falha 17.



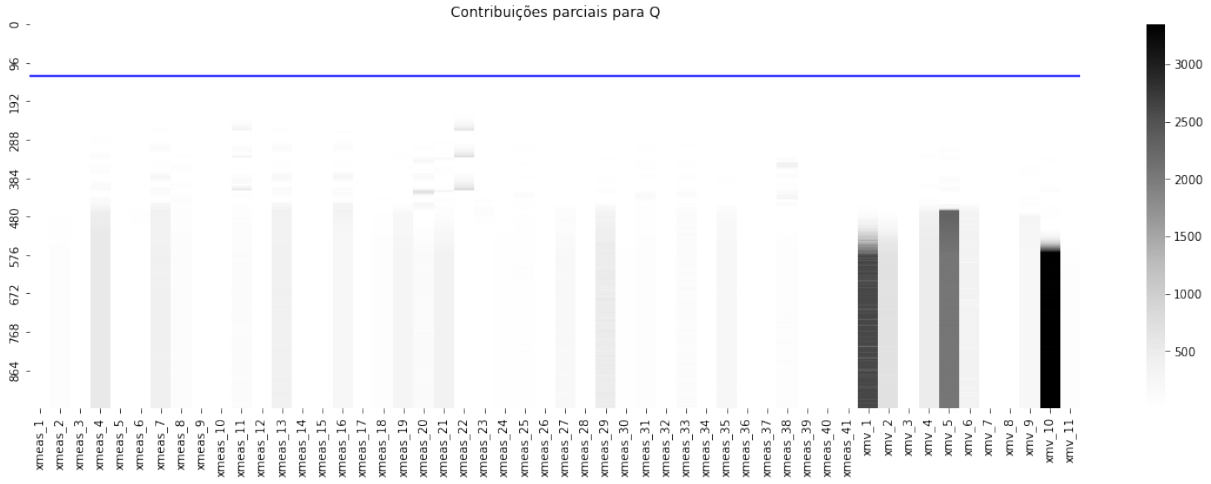
Fonte: Autor, 2022.

Figura 45 – PCA aplicado à falha 18.



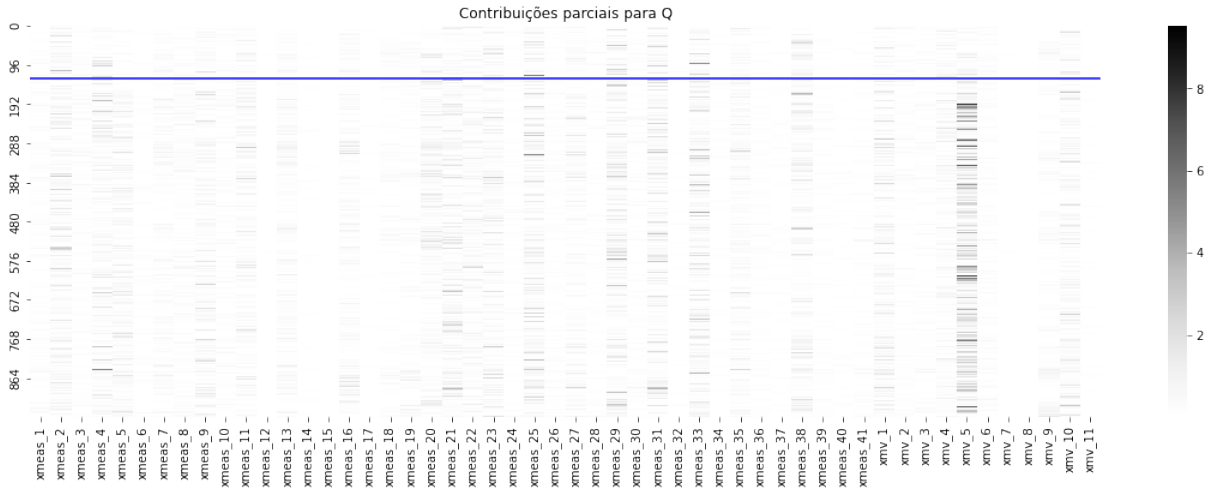
Fonte: Autor, 2022.

Figura 46 – Contribuições parciais Q na falha 18.



Fonte: Autor, 2022.

Figura 47 – Contribuições parciais Q na falha 19.



Fonte: Autor, 2022.