Explicabilidade de variáveis latentes em autoencoders

Kenzo Takayasu Iwasaki[a], Guilherme Augusto Silva de Souza[a*], Song Won Park[a]

[a]Escola Politécnica da Universidade de São Paulo, São Paulo – SP, Brasil
**guilherme_souza@usp.br*

RESUMO

Autoencoders são uma das soluções de aprendizado profundo disponíveis como ferramentas de detecção e diagnóstico de falhas, uma área fundamental da engenharia de sistemas em processos. Com discussões ricas sobre topologia e arquitetura de redes, bem como estratégias de regularização de treinamento a fim de melhorar a explicabilidade dessas redes, não há uma análise profunda do espaço latente dessas redes para essa finalidade. Esse trabalho propõe o estudo das variáveis latentes para melhorar a capacidade diagnóstica do *autoencoder*, que comumente é treinado para ganhar capacidade de reconstrução sem o controle do fluxo de informação na rede. Esse estudo é feito duas redes de topologias distintas quanto ao uso de unidades LSTM e lineares, e controle do fluxo de informação na rede por meio de máscaras de gradientes aplicadas durante o treinamento. Ambas as redes demonstraram a capacidade de distinguir entre uma falha grosseira e uma falha autocompensada no processo Tennessee Eastman.

Palavras-chave: Detecção e diagnóstico de falhas, processo Tennessee Eastman, autoencoders, long-short term memory, machine learning.

1. Motivação

Detecção e diagnóstico de falhas operacionais em uma planta industrial química, ou análise multivariada, é uma estratégia de monitoramento de medidores de uma planta química responsável por avaliar em tempo real os dados de medidas de uma planta química em busca de anomalias (detecção) e identificar qual é o tipo de anomalia que ocorre (diagnóstico). Alves e Nascimento (2004) propuseram o uso da análise de componentes principais para a detecção de *outliers* e erros sistemáticos em uma planta de isopreno. Almeida e Park (2009) propõem o uso de modelos ocultos de Markov para o monitoramento de depósito de cinzas em caldeiras. Parágrafo sobre detecção de falha. Garcia e Munaro (2016) propõem análise de componentes principais (PCA, do inglês) seguida da construção de um mapa causal para isolar a falha tanto no processo Tennessee Eastman (TE) quanto em uma planta termoelétrica de produção de energia. Já D'Angelo et al. (2016) propõe uma nova abordagem usando estatística Bayesiana e fuzzy para detectar a falha, seguido do uso combinado de algoritmos de aprendizado de máquina para classificar a falha detectada. Zuqui et al. (2016) propõe uma combinação de análise de componentes principais, estatística de Hotelling e descritores baseados na distância de Mahalanobis para classificar os diferentes modos operacionais de um reator tanque continuamente agitado, uso de aprendizado supervisado para o diagnóstico de estados que não se aproximem dessas regiões de operação, e aprendizado contínuo para esse sistema de diagnóstico.

O uso de aprendizado de máquina para detecção e diagnóstico de falhas não é recente. Com os avanços constantes de arquitetura de rede – quanto a unidades, profundidade de rede e transferência de aprendizado – existe uma multitude de trabalhos desenvolvidos aliando detecção e diagnóstico de falhas com soluções de inteligência artificial. Trabalhos que envolvem especificamente o processo TE também são numerosos, mostrando a variedade de soluções disponíveis para estudo e aplicação na indústria. Boldt, Rauber e Varejão (2014) propuseram o uso de uma *extreme learning machine*, consistindo de uma rede rasa (com uma camada oculta de *perceptrons* pesos aleatórios e a camada de saída com uma combinação linear dos sinais oriundos da camada oculta a ser ajustada por uma minimização dos erros quadráticos) para o diagnóstico automático de falhas do processo TE. Oliveira et al. (2017) fizeram o uso de redes neurais sem peso, com a vantagem de não trabalhar com resíduos ou evitar novos treinamentos da rede, para a detecção e diagnóstico de falhas no processo TE. Xavier e Seixas (2018) utilizaram LSTM (*long short-term memory*) para a detecção de falhas no mesmo processo.

Com o ganho de poder computacional com o advento da aceleração por unidade de processamento gráfico (GPU), uso de redes profundas começou a ser favorecido. Souza, Souza Jr. E Silva (2021) investigaram o potencial de três topologias de redes neurais convolucionais para a detecção e diagnóstico de falhas na operação

do processo TE. As variações de topologia nesse trabalho vêm do questionamento da necessidade de camadas totalmente conectadas. Além disso, houve uma redução de 80% no tempo de treinamento da rede convolucional com topologia proposta, diferente da topologia de rede convolucional com camadas totalmente conectadas. Kanno e Kaneko (2022) propõem uma rede convolucional profunda com deconvolução associada a um autoencoder profundo para detecção e diagnóstico de falhas do processo TE. Zhang e Xu (2023) propõem uma combinação de uma rede convolucional (CNN, do inglês) com uma rede *long short-term memory* (LSTM) e um mecanismo de atenção que é capaz de detectar e diagnosticar a falha na operação do processo TE. Ewuzie et al. (2025) compararam diversas abordagens de aprendizado de máquina (autoencoders, LSTM, CNN) enquanto usaram técnicas de redução de dimensionalidade dos dados (PCA, *support vector machines* e *random forest*) para melhorar a performance das redes na detecção e diagnóstico de falhas no processo TE – usando o *score* F1 (Tharwat, 2020) para comparar a performance das soluções implementadas.

Uma conseqüência do uso de redes profundas é a perda de interpretabilidade. Pensando em uma rede de detecção de falhas por erro de reconstrução, não se pode inferir muito além da contribuição de cada sinal reconstruído para o erro total. Rauber, Boldt e Munaro (2020) propõem uma busca dos principais sinais que contribuem para uma falha no processo TE. Silva et al. (2025) melhoram a explicabilidade de uma rede neural de 500 neurônios ao se incorporar a regularização L1 e L2 durante o treinamento, e um passo de linearização das funções de ativação quando apresentam comportamento quase-linear (ao redor do zero para uma função tangente hiperbólico, no caso do trabalho). Já Melo et al. (2025) propõe o uso de padrões de distâncias matriciais para ganhar interpretabilidade na análise de dados.

Este trabalho propõe uma avaliação das variáveis latentes de um *autoencoder*, a fim de primeiramente entender como é possível ganhar explicabilidade tanto avaliando puramente o espaço latente quanto incorporando uma restrição *hard* como especificar partes do espaço latente para sinais específicos. Na Seção 2, temos uma breve explicação das redes usadas nesse trabalho, enquanto que na Seção 3 temos a aplicação das mesmas, e na Seção 4 algumas considerações finais são feitas com relação aos resultados observados.

2. Metodologia

O processo Tennessee Eastman é um *benchmark* utilizado na engenharia de sistemas em processos de complexidade adequada a processos industriais reais. Com reciclos, malhas de controle e condições de parada, é um processo com 41 variáveis monitoradas e 12 variáveis manipuladas contando também com 21 diferentes falhas adotadas como padrão pela academia nos diferentes trabalhos em que é utilizado. Proposto primeiramente por Downs e Vogel (1993), tinha a finalidade de ser objeto de estudo para controle de planta inteira e controle multivariável. Neste trabalho, faz-se a detecção e diagnóstico de falhas utilizando apenas as variáveis monitoradas, usando o simulador do processo Tennessee Eastman disponibilizado por meio de um *wrapper* em Python desenvolvido por Kitchin (2026).

2.1. *Geração de dados de treinamento, validação, teste e falha*

O *wrapper* permite o uso de diferentes sementes para o gerador de números aleatórios que constroem o ruído observado na simulação. Assim, construiu-se a base de dados de treinamento com quatro simulações de sementes diferentes e mesmo comprimento. Os dados de validação e teste também foram gerados a partir de outras duas simulações com sementes distintas, visando que o treinamento não aprenda o ruído de medida e sim a dinâmica do processo. Essas simulações têm duração de 34 horas com amostragem a cada minuto.

Já os dados de falha foram construídos a partir de sementes distintas e igual freqüência de amostragem, mas com duração menor: 3,4 horas. As falhas de 1 a 6 (degrau na razão de alimentação dos componentes A e C, degrau na composição do componente B na alimentação, degrau na temperatura de alimentação do componente D, degrau na temperatura da água da camisa do reator, degrau na temperatura de alimentação no condensador e na alimentação do componente A) foram introduzidas após a primeira hora de simulação.

Todos os dados foram janelados em bateladas de comprimento 20, para o treinamento da rede com neurônios LSTM, vantajosos para aplicações em séries temporais.

2.2. *Autoencoders*

Os *autoencoders* são redes de aprendizado não-supervisionado responsáveis por reduzir os dados de entrada em uma representação latente, na etapa chamada de codificação ou *encoding*, e em seguida reconstruir os dados alimentados a partir dessa representação latente, na etapa seguinte chamada de decodificação ou *decoding*. No contexto de detecção e diagnóstico de falhas, faz-se o treinamento da rede sobre dados dinâmicos de operação a fim de reconstruir dados em tempo real para detectar pontos em que o processo analisado apresenta

comportamento anômalo a partir da diferença entre o dado alimentado e o dado reconstruído. O *autoencoder* a princípio não é capaz de classificar a falha que está acontecendo em tempo real, necessitando de uma análise diferente para esse fim. Aqui, diferente das abordagens propostas na seção de Motivação, faz-se uma análise do espaço latente em busca de caracterização das falhas a depender de sua ativação dos neurônios da camada latente.

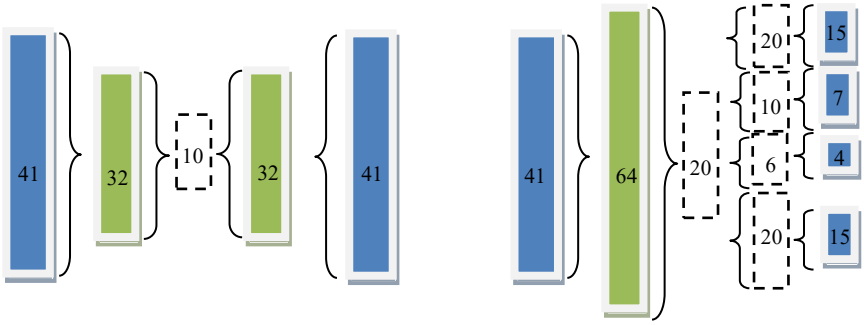
2.2.1. Autoencoder simples

Neste trabalho, propõe-se o uso de um *autoencoder* simples e espelhado, que consiste da camada de entrada para as variáveis alimentadas, uma camada LSTM, uma camada gargalo totalmente conectada e outra camada LSTM, seguida de uma camada de saída para as variáveis reconstruídas. A ilustração das camadas de neurônios está disponível na Figura 1. O número de neurônios utilizados nas camadas LSTM e gargalo são respectivamente 32 e 10. O treinamento dessa rede é feito usando o algoritmo ADAM proposto por Kingma e Ba (2015). O treinamento teve duração limite de 50 épocas com dados de treinamento e validação embaralhados em toda época.

2.2.2. Autoencoder especialista

Propõe-se também um *autoencoder* especialista. O propósito dessa rede é particionar o espaço latente exclusivamente por equipamento. O processo Tennessee Eastman que originalmente consiste de cinco operações unitárias foi separado em quatro partes: reator, separador, compressor e stripper (com condensador). Os sensores dessas partes foram separados e usados para construir uma máscara de gradientes a ser utilizada na construção da função perda do *decoder*.

Figura 1: Grafo da topologia do *autoencoder* simples (à esquerda) e *autoencoder* especialista (à direita). Em azul, neurônios de entrada/saída, em verde neurônios LSTM e branco, neurônios totalmente conectados.



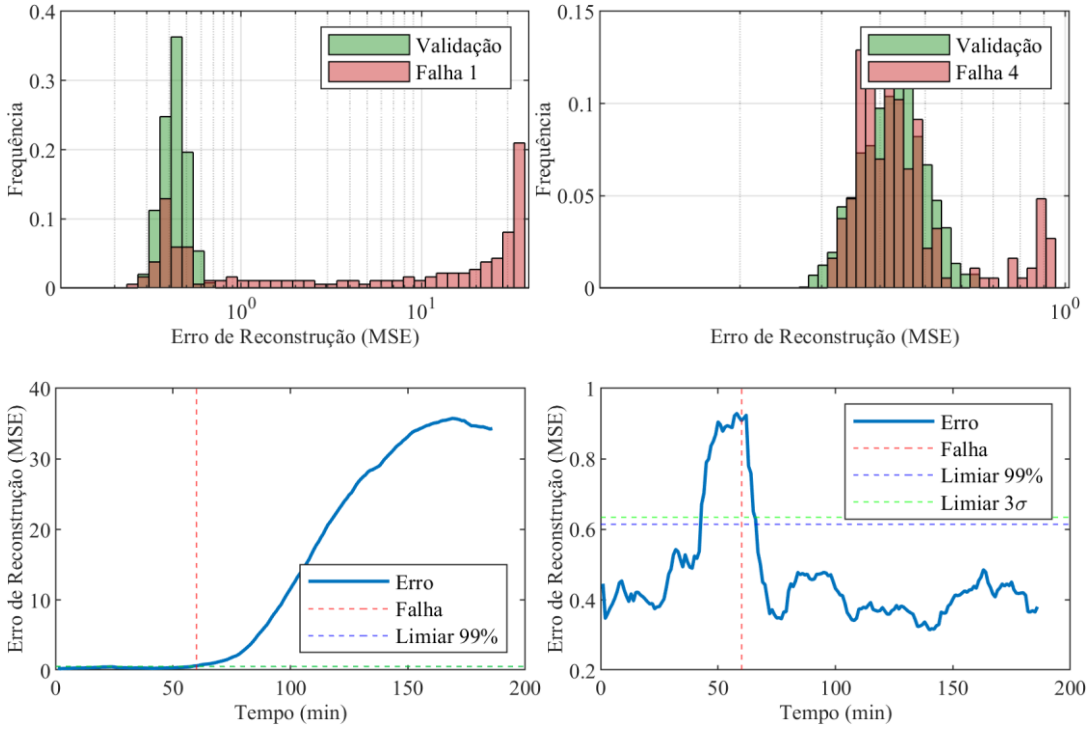
Observando a Figura 1, podemos ver a diferença de simetria na topologia do *autoencoder* simples e o especialista. A topologia da rede especialista não é mais simétrica. Devido ao particionamento do espaço latente, a capacidade de decodificação da rede foi drasticamente reduzida. Para compensar isso, o espaço latente foi expandido para 20 neurônios. A camada LSTM do *encoder* agora tem 64 neurônios, e a camada LSTM do *decoder* foi substituída por uma camada totalmente conectada de quatro partes. O ramo do reator tem uma camada de 20 neurônios totalmente conectados e uma camada de saída de 15 neurônios referente aos sensores dessa parte do processo. O ramo do separador tem uma camada de 10 neurônios totalmente conectados a uma camada de saída de sete neurônios. O ramo do compressor tem uma camada de seis neurônios totalmente conectados seguida de uma camada de quatro neurônios relacionados aos sensores do compressor. Por fim, o ramo do *stripper* tem uma camada de 20 neurônios totalmente conectados a uma camada de saída de 15 neurônios para os sensores deste equipamento. O *decoder* conta no total com 56 neurônios totalmente conectados e 41 neurônios de saída para os sensores das variáveis monitoradas.

O treinamento dessa rede foi feito utilizando o mesmo algoritmo ADAM, porém com 40 épocas de limite e com redução de 80% da taxa de aprendizado a cada 15 épocas, começando em 0,001. A reconstrução da rede usada para cálculo do resíduo foi modificada de forma que uma máscara se aplique aos gradientes do *decoder* de forma que a rede não aprenda a modificar neurônios que não se deseja relacionar com uma parte da planta, enquanto os pesos iniciais dos neurônios sem relação com sensores foram anulados. Essa foi a alternativa a uma regularização L1, que já foi feita na literatura por Silva et al. (2025).

3. Resultados

Primeiramente, avalia-se a capacidade de reconstrução do *autoencoder* simples, avaliando a distribuição dos erros de reconstrução para os dados de validação e para os dados de falha. As falhas 1 e 4 foram selecionadas respectivamente devido a discrepância de seu impacto na planta. A falha 1 causa uma falha de grande magnitude ao longo da planta toda (como fica evidente na Figura 2), mas a falha 4 tem um efeito sutil na planta e é autocompensada pelas malhas de controle ativas na planta.

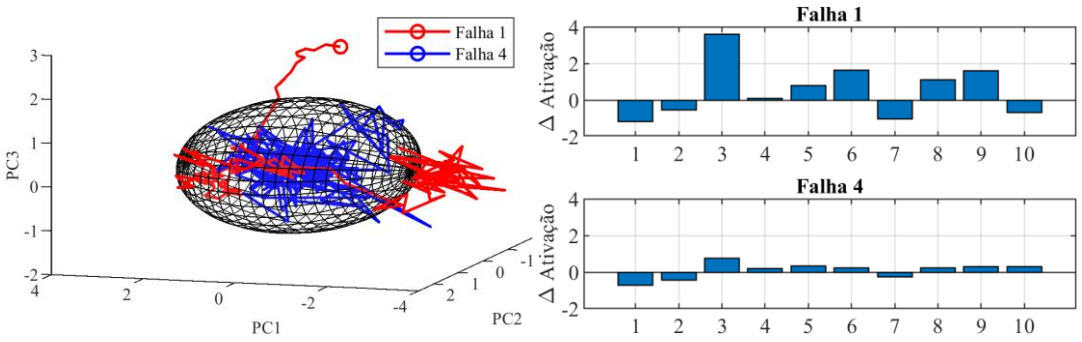
Figura 2: Histogramas de erro de reconstrução do *autoencoder* simples dos dados de validação comparado às falhas 1 e 4, acima, e perfil de erro de reconstrução ao longo do tempo, abaixo.



Utilizou-se dois critérios de detecção de falha, o erro de 99-percentil e o erro 3σ, calculados a partir dos dados de treinamento da rede. Os dados dos histogramas foram normalizados conforme a probabilidade do erro, a fim de normalizar dois conjuntos de pontos com tamanhos distintos, porque os dados de validação têm aproximadamente dez vezes mais amostras do que os dados de falha. Além do comportamento esperado, é possível ver que a depender da falha o *autoencoder* é capaz de detectá-la sem ter uma janela cheia (de comprimento 20) de dados da planta em falha. Essa é uma característica importante quando se interessa no desempenho de detecção da rede, pois aponta o atraso de detecção.

Na Figura 3, podemos visualizar as trajetórias de erro no espaço reduzido por análise de componentes principais. Os três primeiros componentes do espaço reduzido dos dados de treinamento descrevem 62% da informação, e o elipsoide aponta a região de confiança de 95% para dados de treinamento.

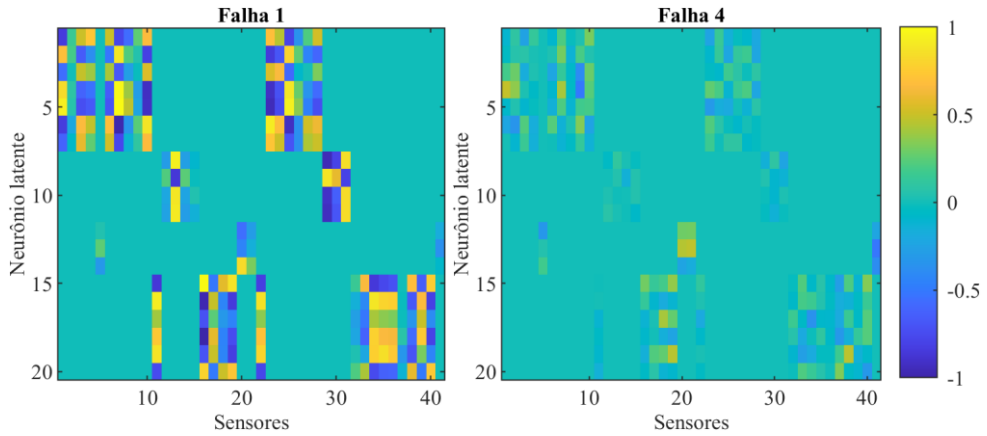
Figura 3: Espaço reduzido dos dados de treinamento com trajetórias das falhas 1 e 4 (esquerda) e assinatura latente de falhas no *autoencoder* simples.



As trajetórias de falha claramente saem do elipsoide, indicando o não-pertencimento parcial da trajetória às condições normais de operação. Como já foi visto no perfil de erro de reconstrução da Figura 2, a malha de controle da planta consegue trazer a planta da condição de falha para a condição normal de operação. Porém, a falha 1 termina sua trajetória fora do intervalo de confiança (indicado pelo marcador). Fazendo uma análise mais profunda dessa rede, avalia-se o espaço latente do *autoencoder* na Figura 3, em que as assinaturas latentes dessas falhas são comparadas. Essa assinatura foi calculada com a diferença entre as ativações médias da falha (a partir dos 100 minutos) e as ativações médias dos dados de treinamento da rede. Por brevidade, apenas as assinaturas latentes das falhas 1 e 4 são apresentadas. Verificou-se que cada falha tem um perfil de ativação média distinto na camada latente do *autoencoder*, o que torna essa abordagem interessante para o uso em visão computacional. Devido à mistura de informação na camada latente, todos os neurônios participam na reconstrução das entradas do *autoencoder*.

Observando agora o *autoencoder* especialista, com separação de neurônios por equipamento, temos a assinatura latente dos neurônios para as falhas 1 e 4 em função dos sensores a serem reconstruídos na Figura 4.

Figura 4: Assinatura latente de falhas no *autoencoder* especialista.



A estratégia de mascarar os gradientes da rede durante seu treinamento funcionou, tanto verificando os pesos das camadas conectadas do decodificador quanto visualizando as ativações para as falhas 1 e 4 da Figura 4. Essa especialização do decodificador por equipamentos entrega uma assinatura latente esparsa, mais fácil de diagnosticar por estratégias como visão computacional.

4. Considerações finais e trabalhos futuros

Os *autoencoders* utilizados nesse trabalho são capazes de distinguir entre diferentes falhas devido ao seu comportamento no espaço latente, tanto para a rede simples quanto a especialista. Uma possível evolução desse trabalho é entender se há perda de capacidade de detecção ou aumento do atraso de detecção, uma vez que o

desempenho das redes implementadas neste trabalho não foi interessante, sendo a maior preocupação a explicabilidade das redes desenvolvidas.

Um possível desenvolvimento para este trabalho é a avaliação do desempenho da rede especialista desenvolvida, em especial avaliar se o ganho de explicabilidade altera de alguma forma a capacidade e o atraso de detecção. Outros possíveis desdobramentos envolvem avaliar a assinatura latente em tempo real, a fim de avaliar se essa assinatura apresenta um comportamento dinâmico.

Agradecimentos: O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. Processo 88887.988471/2024-00.

Referências

R. M. B. Alves e C. A. O. Nascimento: Analysis and Detection of Outliers and Systematic Errors in Industrial Plant Data, Chemical Engineering Communications (194), p. 382-397, 2007.
G. M. Almeida e S. W. Park: Ash Deposits Monitoring in a Convective Heat Transfer Section of a Craft Recovery Boiler, Computer Aided Chemical Engineering (27), p. 1467 – 1472, 2009.
F. A. Boldt, T. W. Rauber e F. M. Varejão: Evaluation of the Extreme Learning Machine for Automatic Fault Diagnosis of the Tennessee Eastman Chemical Process, IECON 2014 – 40th Annual Conference of the IEEE Industrial Electronics Society, p. 2551-2557, 2014.
D. P. Kingma e J. L. Ba: Adam: A Method for Stochastic Optimization. Proceedings of International Conference on Learning Representations 2015. 2015.
G. M. Garcia e C. J. Munaro: Isolation of plant-wide faults using causality detection methods, IFAC-PapersOnLine (49)-7, p. 13-18, 2016.
M. F. S. V. D'Angelo e R. M. Palhares e M. C. O. Camargos Filho e R. D. Maia e J. B. Mendes e P. Y. Ekel: A new fault classification approach applied to Tennesse Eastman benchmark process. Applied Soft Computing (49), p. 676-686, 2016.
J. C. M. Oliveira, K. V. Pontes, I. Sartori, M. Embiruçu: Fault Detection and Diagnosis in Dynamic Systems using Weightless Neural Networks, Expert Systems With Applications (84), p. 200-219, 2017.
G. M. Xavier e J. M. Seixas: Fault detection and diagnostics in a chemical process using long short-term memory recurrent neural network. 2018 International Joint Conference on Neural Networks (IJCNN), p. 1-8, 2018.
T. W. Rauber e F. A. Boldt e C. J. Munaro: Feature selection for multivariate contribution analysis in fault detection and isolation, Journal of the Franklin Institute (357), p. 6294-6320, 2020.
A. Tharwat: Classification assessment methods, Applied Computing and Informatics (17), p. 168 – 192, 2020.
A. C. O. Souza, M. B. Souza Jr., F. V. Silva: Exploring the Potential of Fully Convolutional Neural Networks for FDD of a Chemical Process, Computer Aided Chemical Engineering (49), p.1621-1626, 2022.
Y. Kanno e H. Kaneko: Deep Convolutional Neural Network with Deconvolution and a Deep Autoencoder for Fault Detection and Diagnosis. ACS Omega (7), p. 2458-2466, 2022.
A. Melo e F. F. Fadel e M. M. Câmara e J. C. Pinto: Distance Matrix Patterns for Visual and Interpretable Process Data Analytics, Industrial & Engineering Chemical Research (62), p. 13889-13901, 2025.
H. Zhang e B. Xu: Chemical Process Fault Diagnosis Method Based on Deep Learning, J. Phys.: Conf. Ser. (2637), 2023.
R. N. Ewuzie e S. Gunasekaran e Z. Ahmad e N. M. Nor: Design and implementation of deep learning-based framework for multi-class fault diagnosis in complex chemical process systems, Engineering Applications of Artificial Intelligence (162), 2025.
P. Silva e Y. Ferreira e I. Lima e E. Vakkilainen e G. Almeida e A. Braga: Improving interpretability for fault diagnosis in complex chemical processes using combined regularization and linearization in neural network modeling, 2025 IEEE Symposium on Trustworthy, Explainable and Responsible Computational Intelligence, p. 1-7, 2025.
J. Kitchin. Tennessee Eastman Process Simulator, GitHub, https://github.com/jkitchin/tennessee-eastman-profbraatz/, 2026.