

# Identificação de Falhas de uma Planta de Nível Piloto Baseada em Rede Neural LSTM

Emerson Vilar de Oliveira <sup>\*,\*\*</sup> Mailson Ribeiro Santos <sup>\*</sup>  
Yuri Thomas Pinheiro Nunes <sup>\*</sup> Luiz Affonso Guedes <sup>\*</sup>

<sup>\*</sup> Departamento de Engenharia de Computação e Automação,  
Universidade Federal do Rio Grande do Norte, RN.

<sup>\*\*</sup> (e-mail: [evilaroliveira@gmail.com](mailto:evilaroliveira@gmail.com))

**Abstract:** Data-driven fault classification in industrial context is an expanding field and widely discussed in literature. Artificial Neural Networks, specially Recurrent Neural Networks, are powerful models used for data-driven fault classification since they transports information through time instants. This work evaluates a Long Short Term Memory based model previously proposed in accurately classifying faults in data from a non-simulated process. Furthermore, the hyper parameter space will be explored with the goal of finding a model with optimized results. Through the tests, a final classification accuracy of 91.49% was obtained for faults from a pilot level plant.

**Resumo:** A classificação das falhas industriais utilizando abordagens orientadas a dados é um campo crescente e amplamente discutido na literatura. As Redes Neurais Artificiais, em especial as Redes Neurais Recorrentes se demonstraram fortes aliadas a esse tipo de abordagem, por sua capacidade de transportar informações contidas no histórico das séries temporais ao longo dos instantes de tempo. Aqui será avaliada a capacidade de um modelo *Long Short Term Memory* previamente proposto em classificar corretamente as falhas em dados gerados obtidos de um processo não simulado. Posteriormente, será testada uma variação dos hiperparâmetros com o objetivo de encontrar um conjunto que apresente resultados otimizados. Através dos testes foi obtido uma acurácia final de 91,49% na classificação de falhas de uma planta de nível piloto.

**Keywords:** Industrial systems; process monitoring; faults diagnosis; long short term memory; pilot plant.

**Palavras-chaves:** Sistemas industriais; monitoramento de processos; diagnóstico de falhas; *long short term memory*; planta piloto.

## 1. INTRODUÇÃO

Devido à necessidade de se melhorar a segurança operacional no ambiente industrial, diversas abordagens de detecção e identificação de falhas (*Fault Detection and Identification* - FDI) foram desenvolvidas e aplicadas, com o objetivo de auxiliar os operadores no controle e observação dos processos industriais (Yu et al., 2015; Zhou et al., 2004; Verron et al., 2010). Dentre estas abordagens, destacam-se as metodologias com base em técnicas de aprendizagem de máquina (Lin et al., 2018; Chen et al., 2015; Liu et al., 2018).

O diagnóstico de falhas é uma importante atividade na prevenção de acidentes, então, um bom sistema de FDI pode ser considerado crítico para o funcionamento desejado do processo. Assim, é necessário avaliar e validar as soluções antes delas serem implementadas nos processos industriais. Para isto, utilizam-se de estratégias de simulação e implementação em protótipos de escala reduzida. Em termos de soluções baseadas em simulação, o *benchmark Tennessee Eastman Process* (TEP) é amplamente

utilizado na literatura (Yin et al., 2012; D'Angelo et al., 2016; Gao and Hou, 2016; Basha et al., 2020). O TEP foi desenvolvido pela *Eastman Chemical Company* com o objetivo de disponibilizar um simulador baseado em um processo químico real para avaliar o controle de processos e métodos de monitoramento (Downs and Vogel, 1993). No uso de dados advindos de processos reais, o *benchmarks DAMADICS (Development and Application of Methods for Actuator Diagnosis in Industrial Control Systems)* tem boa aceitação e utilização na literatura. Além da possibilidade de se gerar dados simulados, este *benchmark* também disponibiliza um conjunto de dados coletados de um processo de evaporação de água de uma fábrica de açúcar polonesa (Bartyś et al., 2006).

Apesar de os *benchmarks* TEP e DAMADICS serem amplamente utilizados para avaliações de desempenho de abordagens de FDI, normalmente os dados aplicados nestas avaliações são obtidos por simulação computacional e os dados reais disponíveis no *benchmark* DAMADICS são extraídos de um sistema simples. Por outro lado, a avaliação do desempenho de determinada abordagem de FDI em um sistema real piloto é um importante passo antes da aplicação em um sistema real, visto que o sistema piloto pode representar com mais fidelidade

\* O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001

as situações encontradas em operação real de processos industriais, como várias fontes de ruídos, não-linearidades e outras variáveis exógenas que normalmente não podem ser simuladas com acurácia.

Assim, tendo em consideração a importância da análise de um método de FDI em dados gerados por um sistema mais próximo dos reais. Neste artigo será apresentado um conjunto de análises feitas em um grupo de falhas obtidas a partir de uma planta piloto de nível. Será utilizada uma abordagem baseada em uma rede neural recorrente do tipo LSTM. As redes neurais recorrentes têm a característica de ponderar novas informações por informações já armazenadas anteriormente na máquina, denotando um sentido de memória (Coppin, 2004). Esse tipo específico de rede foi escolhida devido a sua capacidade de transportar informações por um período maior comparado a uma rede neural recorrente comum, o que é desejado no trato de sinais que possuem correlação temporal (Hochreiter and Schmidhuber, 1997). Além da avaliação da metodologia de FDI em dados colhidos de uma planta piloto, as análises feitas consideram a capacidade do modelo classificar corretamente falhas diferentes, mas que fazem parte de um mesmo grupo e possuem comportamentos semelhantes. Posteriormente, este estudo mostra como o comportamento da acurácia de validação se modifica em relação à variação de parâmetros específicos, com o objetivo de propor uma maneira de seleção desses parâmetros, baseado em um teste exaustivo. A escolha desses parâmetros é realizada seguindo alguns critérios estabelecidos.

Na Seção 2 são abordados os conceitos de isolamento de falhas e de Redes Neurais LSTM, além de alguns trabalhos relacionados. Na Seção 3 é feita uma descrição mais detalhada da metodologia para classificação de falhas. A Seção 4 descreve a planta piloto e o conjunto de dados de falhas geradas para validação do método. Seguindo para a Seção 5, são descritos os procedimentos que o conjunto de falhas serão submetidos para se avaliar os pontos previamente expostos, enquanto na Seção 6, é relatado os resultados obtidos e uma análise do que é observável dos experimentos. Por fim, as conclusões e considerações finais deste trabalho são feitas na Seção 7.

## 2. FUNDAMENTAÇÃO TEÓRICA

Nesta seção são apresentados os conceitos sobre detecção e identificação de falhas em processos industriais, aspectos sobre a área de Aprendizado de Máquina com o enfoque em Redes Neurais do tipo LSTM e uma breve análise sobre os trabalhos que utilizam técnicas de redes neurais no contexto de FDI.

### 2.1 Detecção e Identificação de Falhas

Nas últimas décadas o diagnóstico de falhas, também conhecido como Detecção e Identificação de Falhas (em inglês, *Fault Detection and Identification* - FDI), tornou-se um importante campo de estudo. Entretanto, FDI ganhou mais atenção nos últimos anos devido à complexidade atual dos processos industriais (Precup et al., 2015).

A tarefa de FDI pode ser caracterizada como um problema de classificação em dois estágios, detecção e identificação. A detecção de falha é o estágio onde é identificado se o

sistema trabalha em modo normal ou em modo anormal (defeituoso) de operação. Entretanto, neste estágio, informações sobre as falhas como localização e intensidade não são consideradas. Assim, a fase de detecção de falhas pode ser tratada como um classificador de apenas duas classes, capaz de distinguir se as amostras de dados coletados são determinadas como operação "normal" ou "anormal". Identificação de falhas, por outro lado, refere-se à determinação do tipo, localização e instante da detecção dos diversos tipos de falhas que um processo industrial pode ser sujeito (Precup et al., 2015).

Os métodos de FDI podem ser divididos em três grupos: métodos baseados em modelos quantitativos, métodos baseados em modelos qualitativos e métodos baseados em histórico de processo (Venkatasubramanian et al., 2003). Os métodos baseados em modelos quantitativos e qualitativos são geralmente desenvolvidos com base no conhecimento físico do processo. Em modelos quantitativos esse entendimento é expresso como funções matemáticas entre as entradas e saídas do sistema. Em contraste, nas equações do modelo qualitativo essas relações são expressas em funções qualitativas centradas em diferentes unidades do processo. Por outro lado, as abordagens baseadas em histórico de processo possuem apenas a disponibilidade de dados das variáveis do processo em operação. As redes neurais artificiais estão classificadas dentro destes tipos de métodos. Existem diferentes maneiras pelas quais esses dados podem ser transformados e apresentado como conhecimento *a priori* a um sistema de diagnóstico de falhas. Isso é conhecido como o processo de extração de recursos dos dados históricos do processo e é feito para facilitar o diagnóstico *a posteriori*. Esse processo de extração pode ocorrer principalmente como extração quantitativa ou qualitativa de características (Venkatasubramanian et al., 2003).

### 2.2 Redes Neurais LSTM

A LSTM (*Long Short Term Memory*) é um tipo de *Recurrent Neural Network* (RNN) proposta por Hochreiter and Schmidhuber (1997) que introduz o uso de uma arquitetura composta de estados e funções de ativação. A inserção desses componentes tem como objetivo aumentar a robustez dessas redes aos problemas de *vanish/blow up* dos gradientes, presentes nas RNN tradicionais (Hochreiter and Schmidhuber, 1997). Esses problemas ocasionam aumento ou diminuição excessiva dos gradientes, fazendo com que a capacidade de aprendizagem e armazenamento de informação da rede seja prejudicada.

As estruturas utilizadas nas células LSTM são denominadas como, *Cell State*, *Cell Update*, *Forgot Gate*, *Input Gate* e *Output Gate*, cujas suas funcionalidades são descritas a seguir:

- *Cell State*: ponderar pela nova entrada as informações que serão propagadas para o próximo instante de tempo.
- *Cell Update*: realiza a união das informações dos estados passados, ponderados pelas novas entradas e pelo que deve ser esquecido (*Forgot Gate*) pela rede.
- *Forgot Gate*: possibilita que informações passadas possam ser desconsideradas ("esquecimento" de in-

formação). No caso, o decaimento dos gradientes são dependentes da estrutura *Forgot Gate*.

- *Input Gate*: É responsável por decidir o quanto de informação será inserida no *Cell State*.
- *Output Gate*: define quais partes do *Cell State* serão enviados para a saída da rede.

A saída de uma célula LSTM consiste em um *Cell State* e um *Hidden State*. O *hidden state* trata-se de um propagador interno de informação presente entre cada neurônio em uma camada da rede LSTM. As operações feitas na rede através de instantes de tempo são definidas pelo conjunto de equações (1) (Greff et al., 2016).

$$\begin{aligned}
i_t &= \sigma(W_{ii}x_t + b_{ii} + W_{hi}h_{(h-1)} + b_{hi}) \\
f_t &= \sigma(W_{if}x_t + b_{if} + W_{hf}h_{(h-1)} + b_{hf}) \\
g_t &= \tanh(W_{ig}x_t + b_{ig} + W_{hg}h_{(h-1)} + b_{hg}) \\
o_t &= \sigma(W_{io}x_t + b_{io} + W_{ho}h_{(h-1)} + b_{ho}) \\
c_t &= f_t * c_{(t-1)} + i_g * g_t \\
h_t &= o_t * \tanh(c_t)
\end{aligned} \tag{1}$$

Sendo  $h_t$  o *hidden state* no instante de tempo  $t$ ,  $c_t$  o *cell state* no instante de tempo  $t$ ,  $x_t$  o *input* no instante de tempo  $t$ ,  $h_{(t-1)}$  o *hidden state* no valor de tempo  $t - 1$  e  $i_t$ ,  $f_t$ ,  $g_t$  e  $o_t$  são os *input*, *forgot*, *cell* e *output gates* respectivamente. A função  $\sigma$  é a função de ativação Sigmoid.  $W_i$  é uma matriz de pesos de dimensão  $H \times X$ , sendo  $H$  a quantidade de neurônios presentes em uma camada da rede e  $X$  a dimensão do vetor de entrada da rede  $x$ .  $W_h$  é a matriz referente aos pesos dos estados ocultos da rede, com dimensão  $H \times H$ . A variável  $b$  é o *bias* adicionado ao cálculo.

Em Graves and Schmidhuber (2005) foi proposto um recálculo dos gradientes para melhoria de convergência das redes LSTM. Foi mostrado melhores resultados no armazenamento de informações com o uso do cálculo completo dos gradientes, sem nenhum truncamento. Este tipo de mudança caracteriza a **Vanilla LSTM**, que foi difundida e atualmente é uma variação da LSTM amplamente adotada (Greff et al., 2016).

### 2.3 Trabalhos Baseados em Rede Neural para Detecção de Falhas

Entre os pioneiros que utilizaram redes neurais em FDI podemos destacar modelos mais básicos de redes neurais, como as *Multilayer Perceptrons* (MLP) (Sorsa et al., 1991; Ayoubi, 1994). Esse tipo de rede também já foi utilizada em conjunto a outros tipos de técnicas de aprendizado de máquina em Jedliński and Jonak (2015). Combinar diferentes técnicas se demonstra uma tendência entre as soluções para o problema de FDI, onde um procedimento é utilizado para a extração de características e outro para classificação.

Nesta mesma linha, alguns trabalhos utilizam redes LSTM combinadas a outras técnicas como, *Support Vector Machine* (SVM) (Zhang et al., 2017) e *Convolutional Neural Networks* (CNN) (He et al., 2019; Dey et al., 2017). Em Li et al. (2018), uma rede neural LSTM foi utilizada em conjunto com a técnica de processamento de sinais *Wavelet Transform* (WT) (Li et al., 2018). Uma solução que utiliza *Spectral Kurtosis* (SK) e *Extreme Machine Learning* (EML) é apresentada em Udmale and Singh (2019).

## 3. METODOLOGIA

A metodologia utilizará a abordagem proposta por Zhao et al. (2018), que faz uso de uma rede LSTM para classificação de falhas geradas através do simulador de processos industriais TEP (Downs and Vogel, 1993). No caso, a rede neural LSTM é utilizada em conjunto com uma técnica de reparametrização chamada de *Batch Normalization* (BN) (Ioffe and Szegedy, 2015). O objetivo do uso da BN é melhorar a convergência do modelo nas iterações de treino. Essa técnica padroniza as funções de ativação usando de estimativas da média  $\hat{E}(\mathbf{h})$  e desvio padrão  $\hat{V}ar(\mathbf{h})$  para cada camada da rede. Assim, há a normalização no termo recorrente  $W_{hi}h_{(t-1)}$  e não no termo atual  $W_{ii}x_t$ . A equação (2) define a *Batch Normalization*.

$$BN(\mathbf{h}; \eta, \rho) = \rho + \eta \odot \frac{\mathbf{h} - \hat{E}(\mathbf{h})}{\sqrt{\hat{V}ar(\mathbf{h}) + \epsilon}} \tag{2}$$

Sendo  $\mathbf{h}$  o vetor de características a serem normalizadas,  $\rho$  e  $\eta$  os parâmetros que determinam a média e o desvio padrão das características normalizadas e  $\epsilon$  o parâmetro de regularização. O símbolo  $\odot$  representa a multiplicação, elemento a elemento.

Para a rede neural LSTM, as saídas podem ser obtidas a cada instante de tempo. Aqui, a entrada da rede terá um tamanho  $J$ , que corresponde ao instante atual de tempo e  $J - 1$  regressores, enquanto a saída terá dimensão igual à quantidade de classes para a classificação. Após o processamento, a saída da rede é inserida em uma camada totalmente conectada. Esta última camada possui uma função de ativação *LogSoftmax* que é definida na equação (3).

$$LogSoftmax(y_i) = \log \left( \frac{\exp(y_i)}{\sum_j \exp(y_j)} \right) \tag{3}$$

Sendo  $x_i$  a  $i$ -ésima saída da rede e  $x_j$  são as demais saídas para o mesmo instante de tempo.

A função de perda escolhida para cálculo do erro de classificação no treinamento foi a *Negative Log Likelihood Loss* (NLLLoss). Sendo  $c_1, c_2, \dots, c_N$  um grupo de classes  $C$  de falhas e  $f(\cdot)$  a função indicadora, que representa o modelo, de modo que:

$$f(s) = \begin{cases} 1, & \text{se } s \text{ avaliado como verdadeiro} \\ 0, & \text{se } s \text{ avaliado como falso} \end{cases} \tag{4}$$

Onde  $s$  é uma expressão booleana a ser avaliada.

A função de perda é definida tendo como argumento a quintupla  $I = (W_h, W_x, W_{yh}, b, b_y)$  e é descrita da seguinte maneira (Zhao et al., 2018) (conjunto equações (5)):

$$\begin{aligned}
d(\mathbf{x}_t, c_j) &= f(\mathbf{x}_t \in c_j) \\
\mathbf{Loss}(I) &= -\frac{1}{T - J + 1} \sum_{t=1}^{T-J+1} \sum_{j=1}^C d(\mathbf{x}_t, c_j) y_{t,j}
\end{aligned} \tag{5}$$

Onde  $\mathbf{y}_t$  é a saída da rede para o instante de tempo  $t$  e a função indicadora é utilizada para contabilizar os acertos baseados na classe esperada da entrada  $\mathbf{x}_t$ .

A proposta de Zhao et al. (2018) não faz uso de "extração de característica" + "classificação" para o problema de FDI. O modelo recebe diretamente o conjunto multivariável de falhas e tem como saída os *scores* para cada classe de

falha. Isso é feito utilizando janelas de tempo que contêm o instante de tempo atual e passados do processo como entradas. As entradas alimentam a rede LSTM associada à BN, que aplica a normalização aos estados ocultos. Após essas operações, a saída da rede LSTM alimenta uma camada totalmente conectada. A saída da última camada da rede é redimensionada aplicando a função *LogSoftmax*, transformando as saídas em probabilidades de pertinência à cada classe. Para otimização dos gradientes, foi escolhido o algoritmo Adam (Kingma and Ba, 2014).

O código com a implementação completa do método feito pelos autores está disponível online<sup>1</sup>. Foi utilizado o *framework* para aprendizado de máquina *PyTorch* na linguagem de programação *Python*.

#### 4. PLANTA PILOTO

A planta piloto de controle industrial, desenvolvida pela De Lorenzo do Brasil permite a realização de estudos de controle de processos contínuos, baseado em quatro variáveis: pressão, temperatura, vazão e nível, definidas pelo vetor de entrada  $S = (u, t, f, y)$  (Marins, 2009).

A planta piloto possui um transdutor de sinal físico em elétrico, que é processado pelo controlador lógico programável (CLP); barramento terminal; *software* supervisorio e de aquisição de dados (SCADA), para configuração dos parâmetros e visualização do processo. A planta é composta por: um painel com CLP e sistema elétrico para o controle da planta; dois reservatórios pressurizados, um feito de acrílico, *T1*, e o outro feito de aço inoxidável, *T2*; uma bomba de recirculação centrífuga controlada por um inversor de frequência; um sistema de aquecimento e troca de calor; duas válvulas direcionais, *V1* e *V2*; sensores de temperatura, pressão, vazão e nível (Costa, 2014). A Figura 1 mostra a planta piloto utilizada.

Os dois tanques são conectados por um sistema de tubulação que permite o fluxo entre eles. A planta funciona de maneira que é possível transferir o líquido entre ambos os tanques, estando *T1* posicionado acima de *T2* em relação ao nível do solo. O líquido flui sempre em uma direção: de *T1* para *T2* por gravidade, e de *T2* para *T1* a partir da pressão gerada na bomba centrífuga. A Figura 2 ilustra o diagrama esquemático da planta (Costa, 2014).



Figura 1. Planta piloto. Fonte: Costa (2014)

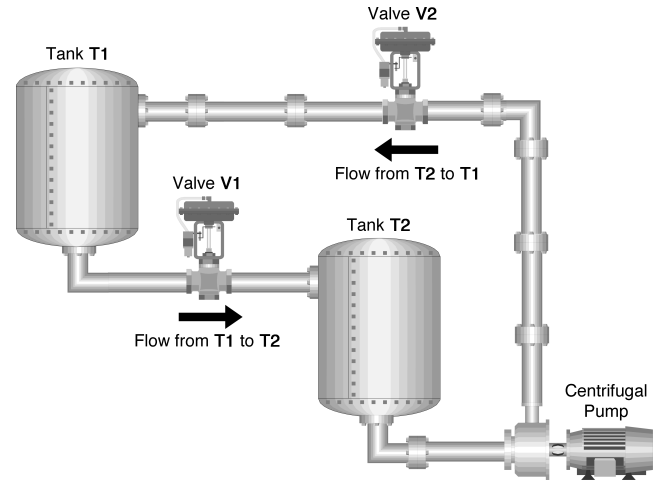


Figura 2. Diagrama esquemático da planta. Fonte: Costa (2014)

Os dados obtidos da planta piloto apresentam diferentes grupos de falhas: Atuador, Sensor, Estrutural e Perturbação. Cada grupo possui níveis de intensidade diferentes para falhas de um mesmo tipo. Na Tabela 1 são apresentadas as falhas e suas descrições.

Tabela 1. Conjunto de falhas geradas.

Falha	Grupo	Descrição
F1	Atuador	+ 2% de <i>off-set</i>
F2		+ 4% de <i>off-set</i>
F3		+ 8% de <i>off-set</i>
F4		- 2% de <i>off-set</i>
F5		- 4% de <i>off-set</i>
F6		- 8% de <i>off-set</i>
F7	Sensor	+ 2% de <i>off-set</i>
F8		+ 4% de <i>off-set</i>
F9		+ 8% de <i>off-set</i>
F10		- 2% de <i>off-set</i>
F11		- 4% de <i>off-set</i>
F12		- 8% de <i>off-set</i>
F13	Estrutural	100% de vazamento do tanque
F14		66% de vazamento do tanque
F15		30% da válvula presa V1
F16		50% da válvula presa V1
F17		85% da válvula presa V1
F18		100% da válvula presa V1
F19		25% da válvula presa V2
F20		50% da válvula presa V2
F21		75% da válvula presa V2
F22	Perturbação	Baixa perturbação
F23		Alta perturbação

No grupo Atuador, existem 6 níveis de *off-set* na bomba +2%, +4%, +8%, -2%, -4% e -8%; no grupo Sensor também existem 6 níveis de *off-set*; no grupo Estrutural existem 2 níveis de abertura de dreno, que simulam fisicamente um vazamento no tanque *T1* e 3 níveis de emperramento para cada uma das válvulas; no grupo Perturbação, existem 2 níveis.

Neste trabalho foram utilizados dados apenas referentes ao controle de nível da planta, onde a planta foi controlada por um controlador *fuzzy* multiestágios, desenvolvido no *software* JFuzZ (Liu et al., 2005). A comunicação entre o Sistema SCADA e o CLP foi realizada através do protocolo OPC (*OLE for Process Control*), que coleta informação dos

<sup>1</sup> [https://github.com/haitaozhao/LSTM\\_fault\\_detection](https://github.com/haitaozhao/LSTM_fault_detection).

valores das variáveis  $nível(y, \text{variável observável})$ , referência ( $r, \text{set-point}$  definido pelo usuário) e  $pressão(u, \text{ sinal de controle na bomba})$ , para formar o vetor  $x = (r, y, u)$ .

## 5. CONFIGURAÇÃO EXPERIMENTAL

Com os dados coletados na Planta Piloto (Seção 4), alguns experimentos foram feitos utilizando a proposta apresentada na Seção 3. Inicialmente é feita uma seleção nas falhas a serem utilizadas nos experimentos. Em seguida uma avaliação da resposta da metodologia a diferentes arquiteturas de rede neural LSTM. Após a análise do comportamento da metodologia, um modelo será selecionado para validação e classificação das falhas.

Na Figura 3 é apresentado o diagrama de blocos que representa a estrutura lógica da montagem dos experimentos. O módulo denominado *classificador*, que corresponde à rede neural LSTM, recebe dados dos sensores no instante atual e  $J - 1$  regressores ( $Z^{-1}$ ) para cada variável, representados respectivamente por  $x$  e  $x_{-1}, x_{-2}, \dots, x_{-J+1}$ . Estes dados servem de base para a classificação, a saída do classificador indica a falha detectada, representadas por  $y_1, y_2, \dots, y_C$ .

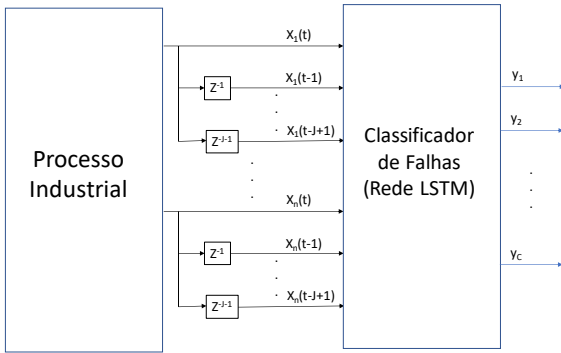


Figura 3. Diagrama representando o método de classificação de falhas.

### 5.1 Seleção das Falhas

Alguns dos distúrbios sobre a operação da planta piloto foram selecionados para a realização dos experimentos. O critério para seleção de falhas foi a média da distância euclidiana entre as variáveis operando sob diferentes falhas. A distância euclidiana para um par de variáveis é definida pela equação (6), onde  $v_{F_a}^i$  e  $v_{F_b}^i$  correspondem a mesma variável para as diferentes falhas (*falha a* e *falha b*). Quanto maior o valor dessa distância, mais o par de falhas em questão é dissimilar.

$$D(v_{F_a}^i, v_{F_b}^i) = \sqrt{\sum_{i=1}^N (v_{F_a}^i - v_{F_b}^i)^2} \quad (6)$$

Onde  $N$  é o tamanho das variáveis em amostras e  $i$  é um índice que identifica qual variável é utilizada.

### 5.2 Seleção da Arquitetura da Rede Neural LSTM

Após a validação das falhas selecionadas para servirem como conjunto de treino da rede neural LSTM, foram

avaliados os impactos dos hiperparâmetros da rede no seu desempenho. Os hiperparâmetros escolhidos para a avaliação foi a quantidade de entradas da rede  $J$  (Seção 3) e a quantidade de neurônios na camada da rede neural LSTM ( $HS$ ). Os valores de  $J$  e  $HS$  testados foram:

- $J = 3, 5, 7, 10$ .
- $HS = 20, 30, 40, 50, 60$ .

A escolha desses valores é baseado no comportamento do conjunto de falhas obtidas através da planta piloto. As falhas geradas a partir deste processo têm um comportamento que varia de forma lenta, não necessitando de uma quantidade alta de amostras passadas para reconhecer possíveis variações de comportamento. Além disso, é desejado avaliar a capacidade das redes LSTM para este tipo de problema, o que levou a utilização de valores não muito altos para a quantidade de neurônios da rede. O espaçamento entre os valores dos hiperparâmetros foram escolhidos visando a redução da quantidade de arquiteturas de rede possíveis, facilitando a visualização dos resultados. Para cada valor de  $J$  foram testados todos os valores de  $HS$ , totalizando 20 combinações dos hiperparâmetros para a arquitetura da rede do modelo.

A métrica escolhida para avaliar o modelo foi a acurácia de acerto na classificação das falhas. O experimento consiste em treinar 100 modelos com os mesmos hiperparâmetros e extrair a métrica de desempenho obtida por cada um deles. Ao final do experimento, cada arquitetura possui 100 valores de acurácias obtidas dos modelos treinados por 50 épocas cada. Ressalta-se que a escolha de um teste exaustivo de validação para várias combinações de valores dos hiperparâmetros ( $J/HS$ ) visa diminuir a aleatoriedade no processo de treinamento da rede LSTM.

## 6. RESULTADOS E ANÁLISES

Para a seleção de falhas, serão apresentados os valores da métrica de dissimilaridade proposta e quais falhas foram selecionadas para treino e validação, respectivamente. Em relação à avaliação do impacto dos hiperparâmetros sobre o desempenho da solução, serão mostrados gráficos que representem as distribuições obtidas nos testes realizados e qual arquitetura atende aos critérios de seleção.

A Tabela 2 mostra um novo agrupamento das falhas em subgrupos correspondentes a diferentes configurações e o cálculo da métrica de dissimilaridade entre as falhas (subseção 5.1). Os grupos Atuador e Sensor foram divididos em *off-sets* positivos e negativos. O grupo Estrutural foi dividido em vazamentos do tanque, da válvula 1 e da válvula 2. O grupo Perturbação não foi utilizado nos experimentos devido a pouca quantidade de amostras disponíveis em seus conjuntos de dados.

Os valores da coluna Dissimilaridade representam o quanto as falhas presentes na coluna Combinação são diferentes. Com o critério de menor dissimilaridade, as falhas selecionadas para treino são as falhas F01, F04, F07, F10, F13, F17 e F19. As que formam o grupo de validação são as falhas, F03, F05, F08, F11, F14, F17 e F20, respectivamente. Tanto para os testes de dissimilaridade quanto para as avaliações subsequentes, foram usadas 600 amostras de cada falha. Esse valor foi escolhido devido à limações de amostras nos conjuntos de dados das falhas, portanto, a

Tabela 2. Seleção das Falhas.

Sub Grupo	Combinação	Dissimilaridade
Atuador (+2%,+4%,+8%)	F01 ↔ F02	61,30
	F01 ↔ F03	<b>46,45</b>
	F02 ↔ F03	51,89
Atuador (-2%,-4%,-8%)	F04 ↔ F05	<b>82,57</b>
	F04 ↔ F06	114,28
	F05 ↔ F06	156,03
Sensor (+2%,+4%,+8%)	F07 ↔ F08	<b>75,75</b>
	F07 ↔ F09	427,42
	F08 ↔ F08	356,74
Sensor (-2%,-4%,-8%)	F10 ↔ F11	<b>72,98</b>
	F10 ↔ F12	282,43
	F11 ↔ F12	236,93
Estrutural Tanque (100%,66%)	F13 ↔ F14	<b>122,48</b>
Estrutural V1 (30%, 50%, 84%, 100%)	F15 ↔ F16	226,43
	F15 ↔ F17	428,27
	F15 ↔ F18	436,50
	F16 ↔ F17	212,47
	F16 ↔ F18	219,92
	F17 ↔ F18	<b>50,41</b>
Estrutural V2 (25%, 50%, 75%)	F19 ↔ F20	<b>41,15</b>
	F19 ↔ F21	75,15
	F20 ↔ F21	83,50

quantidade de entradas para o modelo pode variar de acordo o tamanho da entrada. A equação (7) mostra o cálculo feito para a quantidade de entradas do modelo.

$$Q_e = 600 - (J - 1) \quad (7)$$

Utilizando as falhas previamente selecionadas, o teste para seleção dos hiperparâmetros para o modelo foi aplicado. As Figuras 4 e 5 mostram através de gráficos do tipo *box plot* o comportamento da acurácia dos modelos a partir da variação da janela de entrada da rede  $J$  e tamanho da camada LSTM ( $HS$ ), respectivamente.

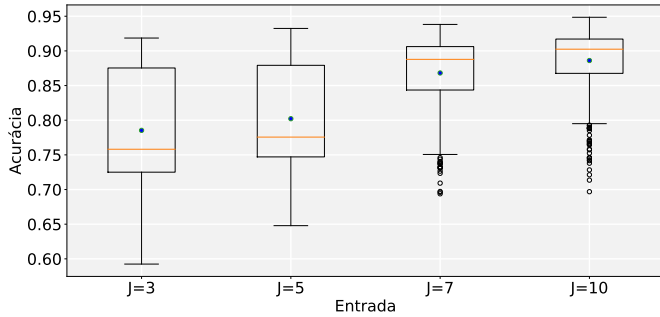


Figura 4. *Box-plot dos valores de acurácia na variação da janela de entrada  $J$ . O gráfico *box-plot* nessa figura destaca as seguintes informações: i) As retas horizontais nas extremidades são os valores máximos e mínimos; ii) A linha superior do *box* delimita o terceiro quartil e a inferior o primeiro quartil, compondo a amplitude interquartílica da distribuição; iii) A linha interior da caixa representa a mediana; iv) O ponto preenchido contém a informação da média e v) Os pontos vazados são valores outliers.*

A partir da Figura 4 é possível observar que os cenários com entradas de tamanho 7 e 10 de amostras das variáveis de processo ( $J = 7$  e  $J = 10$ ) apresentaram comportamento mais consistente em suas distribuições,

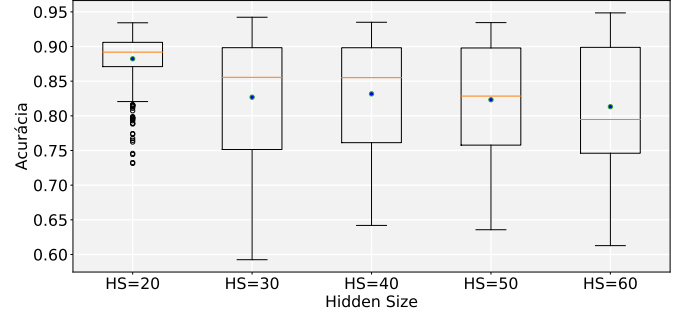


Figura 5. *Box-plot dos valores de acurácia na variação do tamanho da camada oculta da LSTM (*hidden size*).*

considerando todos os tamanhos da rede ( $HS$ ) possíveis. Os valores de acurácia média e mediana são mais elevados e a distância interquartílica menor, indicando uma menor variação no valor da acurácia. Além disso, apresentam valores *outliers* abaixo do *box*, o que significa que baixas acurácias não são comuns. O destaque da Figura 5 é o valor de *hidden size* igual a 20, que possui características semelhantes à avaliação do tamanho das entradas.

A Figura 6 mostra o comportamento de todas as combinações entre os hiperparâmetros da rede neural LSTM de forma isolada a partir do  $J$ . Nesta imagem uma reta no valor de acurácia igual a 0,95 é adicionada, servindo como uma referência visual em relação aos valores observados nos testes.

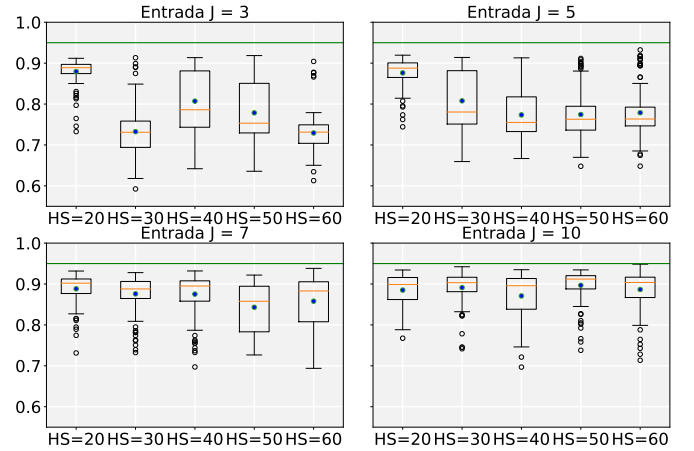


Figura 6. *Box-plot dos valores de acurácia para o teste de combinação dos hiperparâmetros.*

A combinação selecionada dos hiperparâmetros para ser utilizada adiante no estudo, foi a de valores  $J = 7$  e  $HS = 20$ . Essa combinação apresenta valores de acurácia média (0,88), mediana (0,90), variância (0,05) e máxima (0,93) próximos da referência 0,95. Associado a estes valores de acurácia, essa combinação possui uma complexidade computacional mais baixa que combinações como  $J = 10 - HS = 30$  e  $J = 10 - HS = 50$ , onde ambos têm valores de acurácia média **0.89**.

### 6.1 Avaliação do Modelo

Após a seleção da arquitetura ser feita baseada nos critérios previamente apresentados, o modelo treinado foi



avaliado para verificação da acurácia de classificação de forma separada para cada classe. Isso avalia a capacidade do modelo de generalização, ao ser treinado por uma determinada falha e validado a partir de outra que tem comportamento semelhante no mesmo sub grupo. A Figura 7 mostra a matriz de confusão com as acurácias individuais da classificação que foi realizada pela rede neural LSTM com as configurações de hiperparâmetros anteriormente selecionados. No caso, dada a matriz de confusão  $M_C$ , o seu elemento  $M_C(i,j)$  representa quantas vezes o modelo classificou a falha  $i$  como sendo a falha  $j$ .

Classe de Saída	Classe de Entrada							Total	
	F03	F05	F08	F11	F14	F18	F20	Verdadeiro	Falso
F03	568,0 13,66%	0,0 0,00%	0,0 0,00%	0,0 0,00%	26,0 0,63%	0,0 0,00%	0,0 0,00%	95,62%	4,38%
F05	0,0 0,00%	581,0 13,97%	0,0 0,00%	0,0 0,00%	13,0 0,31%	0,0 0,00%	0,0 0,00%	97,81%	2,19%
F08	0,0 0,00%	0,0 0,00%	594,0 14,29%	0,0 0,00%	0,0 0,00%	0,0 0,00%	0,0 0,00%	100,0%	0,00%
F11	0,0 0,00%	0,0 0,00%	0,0 0,00%	594,0 14,29%	0,0 0,00%	0,0 0,00%	0,0 0,00%	100,0%	0,00%
F14	0,0 0,00%	85,0 2,04%	0,0 0,00%	0,0 0,00%	509,0 12,24%	0,0 0,00%	0,0 0,00%	85,69%	14,31%
F18	0,0 0,00%	9,0 0,22%	0,0 0,00%	0,0 0,00%	56,0 1,35%	526,0 12,65%	3,0 0,07%	88,55%	11,45%
F20	3,0 0,07%	0,0 0,00%	0,0 0,00%	0,0 0,00%	36,0 0,87%	123,0 2,96%	432,0 10,39%	72,73%	27,27%
	99,47% 0,53%	86,07% 13,93%	100,0% 0,00%	100,0% 0,00%	79,53% 20,47%	81,05% 18,95%	99,31% 0,69%	91,49% 8,51%	
	F03	F05	F08	F11	F14	F18	F20		
	Identificador da Falha								

Figura 7. Matriz de confusão do teste com as falhas para treino e validação definidas na Seção 5.1.

Então, a partir dos resultados apresentados na Figura 7, verifica-se que para as falhas  $F08$  e  $F11$  o modelo classificou de maneira correta em 100% dos casos, além de não errar em classificar outras falhas como  $F08$  ou  $F11$ . As falhas  $F03$  e  $F05$  também obtiveram bons valores para as suas acurácias, 95,62% e 97,91%, respectivamente. Observando estes quatro resultados, é possível perceber que os grupos de falhas que se encontram na categoria *Atuador* e *Sensor*, possuem um comportamento melhor interpretado pela rede no momento da classificação. Em contrapartida, as falhas  $F14$ ,  $F18$  e  $F20$  tiveram valores mais baixos de acurácia (85,69%, 88,55% e 72,73%) e os maiores índices de erros em relação aos demais. A exemplo, a falha  $F18$  foi erroneamente classificada como falha  $F20$  em 123 amostras de 594 (equação (7)). Isso leva a observar que falhas do grupo *Estrutural* possuem comportamento mais semelhante entre suas falhas, o que gera uma certa dificuldade de separação dos dados para a rede na classificação. De modo geral, o modelo avaliado apresentou uma acurácia final de 91,49%.

## 7. CONCLUSÃO

Neste artigo se analisou o desempenho de uma técnica de FDI baseada em uma rede neural BN-LSTM aplicada

sobre uma planta piloto de nível. Essa planta possui características encontradas em sistemas industriais, como a sua instrumentação (sensores e atuadores) e não-linearidade inerentes ao sistema, o que a aproxima bastante do cenário de um processo industrial real. Com isto, pode-se observar o desempenho e comportamento da abordagem em um processo gerado de forma não simulada, constatando sua possível aplicabilidade em um processo real. A solução apresentou um desempenho aceitável para a classificação das falhas, onde podemos destacar como pontos positivos, a capacidade de classificação de falhas pertencentes a mesma classe, ou sub grupo, como por exemplo, as falhas  $F01$  e  $F03$ , apresentadas em distúrbios no atuador. Além de demonstrar um baixo índice de erro na classificação de falhas que tem origem de tipos diferentes, como falhas no atuador e falhas de vazamento. Outro ponto observado, de forma individual e conjunta, o comportamento das distribuições dos valores de acurácia em relação à variação do tamanho de entrada e tamanho da rede LSTM do modelo. A alteração nestes valores indicaram a necessidade de se encontrar uma combinação específica entre eles para se obter melhores valores de acurácia geral do modelo, tendo em vista o comportamento apresentado pelas distribuições obtidas no teste exaustivo.

Com a arquitetura selecionada a partir dos resultados e critérios obtidos nos experimentos, o modelo selecionado atingiu 91,49% de acurácia final na classificação das falhas de validação, possuindo a acurácia de 100% em falhas específicas. Estes resultados apontam a validade da metodologia em classificar falhas obtidas de fontes não simuladas.

Outros tipos de experimentos podem ser empregados utilizando as mesmas metodologias e dados. Diminuir o número de amostras para treino, com o objetivo de testar a robustez da metodologia a um conjunto limitado de dados. Aplicar outro critério para seleção de falhas e testar outras combinações de dados no treino e validação. Avaliar o impacto do aumento exponencial da rede (*deep learning*) e o quanto isso afeta a acurácia de classificação. Estes são algumas ideias para trabalhos a serem feitos.

## REFERÊNCIAS

- Ayoubi, M. (1994). Nonlinear dynamic systems identification with dynamic neural networks for fault diagnosis in technical processes. In *Proceedings of IEEE International Conference on Systems, Man and Cybernetics*, volume 3, 2120–2125. IEEE.
- Bartyś, M., Patton, R., Syfert, M., [de las Heras], S., and Quevedo, J. (2006). Introduction to the damadics actuator fdi benchmark study. *Control Engineering Practice*, 14(6), 577 – 596. doi: <https://doi.org/10.1016/j.conengprac.2005.06.015>. URL <http://www.sciencedirect.com/science/article/pii/S0967066105001796>. A Benchmark Study of Fault Diagnosis for an Industrial Actuator.
- Basha, N., Sheriff], M.Z., Kravaris, C., Nounou, H., and Nounou, M. (2020). Multiclass data classification using fault detection-based techniques. *Computers & Chemical Engineering*, 136, 106786. doi: <https://doi.org/10.1016/j.compchemeng.2020.106786>. URL <http://www.sciencedirect.com/science/article/pii/S0098135420300090>.

- Chen, Z., Li, C., and Sanchez, R.V. (2015). Gearbox fault identification and classification with convolutional neural networks. *Shock and Vibration*, 2015.
- Coppin, B. (2004). *Artificial Intelligence Illuminated*. Jones and Bartlett illuminated series. Jones and Bartlett Publishers. URL <https://books.google.com.br/books?id=Lc0LqodW28EC>.
- Costa, B.S.J. (2014). *Detecção e Diagnóstico de Falhas Não-supervisionados Baseados em Estimativa de Densidade Recursiva e Classificador Fuzzy Auto-evolutivo*. Ph.D. thesis, UFRN, Natal, RN.
- Dey, D., Chatterjee, B., Dalai, S., Munshi, S., and Chakravorti, S. (2017). A deep learning framework using convolution neural network for classification of impulse fault patterns in transformers with increased accuracy. *IEEE Transactions on Dielectrics and Electrical Insulation*, 24(6), 3894–3897.
- Downs, J. and Vogel, E. (1993). A plant-wide industrial process control problem. *Computers & Chemical Engineering*, 17(3), 245 – 255. doi:[https://doi.org/10.1016/0098-1354\(93\)80018-I](https://doi.org/10.1016/0098-1354(93)80018-I). URL <http://www.sciencedirect.com/science/article/pii/009813549380018I>. Industrial challenge problems in process control.
- D’Angelo, M.F., Palhares, R.M., Filho, M.C.C., Maia, R.D., Mendes, J.B., and Ekel, P.Y. (2016). A new fault classification approach applied to tennessee eastman benchmark process. *Applied Soft Computing*, 49, 676 – 686. doi:<https://doi.org/10.1016/j.asoc.2016.08.040>. URL <http://www.sciencedirect.com/science/article/pii/S1568494616304343>.
- Gao, X. and Hou, J. (2016). An improved svm integrated gs-pca fault diagnosis approach of tennessee eastman process. *Neurocomputing*, 174, 906 – 911. doi:<https://doi.org/10.1016/j.neucom.2015.10.018>. URL <http://www.sciencedirect.com/science/article/pii/S0925231215014757>.
- Graves, A. and Schmidhuber, J. (2005). Framewise phone classification with bidirectional lstm and other neural network architectures. *Neural networks*, 18(5-6), 602–610.
- Greff, K., Srivastava, R.K., Koutník, J., Steunebrink, B.R., and Schmidhuber, J. (2016). Lstm: A search space odyssey. *IEEE transactions on neural networks and learning systems*, 28(10), 2222–2232.
- He, Y., Liu, Y., Shao, S., Zhao, X., Liu, G., Kong, X., and Liu, L. (2019). Application of cnn-lstm in gradual changing fault diagnosis of rod pumping system. *Mathematical Problems in Engineering*, 2019.
- Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735–1780.
- Ioffe, S. and Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.
- Jedliński, L. and Jonak, J. (2015). Early fault detection in gearboxes based on support vector machines and multilayer perceptron with a continuous wavelet transform. *Applied Soft Computing*, 30, 636–641.
- Kingma, D.P. and Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Li, S., Zhao, Y., and Ding, M. (2018). Mobile robot motor bearing fault detection and classification on discrete wavelet transform and lstm network. *Journal of Mechanics in Medicine and Biology*, 18(08), 1840034.
- Lin, J., Su, L., Yan, Y., Sheng, G., Xie, D., and Jiang, X. (2018). Prediction method for power transformer running state based on lstm-dbn network. *Energies*, 11(7), 1880.
- Liu, C., Cheng, G., Chen, X., and Pang, Y. (2018). Planetary gears feature extraction and fault diagnosis method based on vmd and cnn. *Sensors*, 18(5), 1523.
- Liu, J., Lim, K.W., Ho, W.K., Tan, K.C., Tay, A., and Srinivasan, R. (2005). Using the opc standard for real-time process monitoring and control. *IEEE Software*, 22(6), 54–59.
- Marins, A. (2009). *Continuous process workbench. Technical manual*. DeLorenzo Brazil, Brazil.
- Precup, R.E., Angelov, P., Costa, B.S.J., and Sayed-Mouchaweh, M. (2015). An overview on fault diagnosis and nature-inspired optimal control of industrial process applications. *Computers in Industry*, 74, 75 – 94. doi:<https://doi.org/10.1016/j.compind.2015.03.001>. URL <http://www.sciencedirect.com/science/article/pii/S0166361515000469>.
- Sorsa, T., Koivo, H.N., and Koivisto, H. (1991). Neural networks in process fault diagnosis. *IEEE Transactions on systems, man, and cybernetics*, 21(4), 815–825.
- Udmale, S.S. and Singh, S.K. (2019). Application of spectral kurtosis and improved extreme learning machine for bearing fault classification. *IEEE Transactions on Instrumentation and Measurement*, 68(11), 4222–4233.
- Venkatasubramanian, V., Rengaswamy, R., Yin, K., and Kavuri, S.N. (2003). A review of process fault detection and diagnosis: Part i: Quantitative model-based methods. *Computers & Chemical Engineering*, 27(3), 293 – 311. doi:[https://doi.org/10.1016/S0098-1354\(02\)00160-6](https://doi.org/10.1016/S0098-1354(02)00160-6). URL <http://www.sciencedirect.com/science/article/pii/S0098135402001606>.
- Verron, S., Tiplica, T., and Kobi, A. (2010). Fault diagnosis of industrial systems by conditional gaussian network including a distance rejection criterion. *Engineering applications of artificial intelligence*, 23(7), 1229–1235.
- Yin, S., Ding, S.X., Haghani, A., Hao, H., and Zhang, P. (2012). A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark tennessee eastman process. *Journal of Process Control*, 22(9), 1567 – 1581. doi:<https://doi.org/10.1016/j.jprocont.2012.06.009>. URL <http://www.sciencedirect.com/science/article/pii/S0959152412001503>.
- Yu, H., Khan, F., and Garaniya, V. (2015). Nonlinear gaussian belief network based fault diagnosis for industrial processes. *Journal of Process Control*, 35, 178–200.
- Zhang, S., Wang, Y., Liu, M., and Bao, Z. (2017). Data-based line trip fault prediction in power systems using lstm networks and svm. *IEEE Access*, 6, 7675–7686.
- Zhao, H., Sun, S., and Jin, B. (2018). Sequential fault diagnosis based on lstm neural network. *IEEE Access*, 6, 12929–12939.
- Zhou, S., Zhang, J., and Wang, S. (2004). Fault diagnosis in industrial processes using principal component analysis and hidden markov model. In *Proceedings of the 2004 American Control Conference*, volume 6, 5680–5685. IEEE.