

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE  
MINAS GERAIS - *CAMPUS* SABARÁ  
BACHARELADO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Vitoria Rodrigues Moraes

**DETECÇÃO DE FALHAS EM UM SISTEMA REATOR DE TANQUE  
COM AGITAÇÃO CONTÍNUA UTILIZANDO INTELIGÊNCIA  
ARTIFICIAL**

Sabará  
2024

VITORIA RODRIGUES MORAIS

**DETECÇÃO DE FALHAS EM UM SISTEMA REATOR DE TANQUE  
COM AGITAÇÃO CONTÍNUA UTILIZANDO INTELIGÊNCIA  
ARTIFICIAL**

Trabalho de Conclusão de Curso apresentado à banca examinadora do curso de Engenharia de Controle e Automação do Instituto Federal de Educação, Ciência e Tecnologia de Minas Gerais *Campus* Sabará, como parte dos requisitos para obtenção do título de Bacharel em Engenharia de Controle e Automação.

**Orientador:** Profa. Dra. Mariella Maia Quadros

**Coorientador:** Prof. Me. Luiz Guilherme Hilel Drummond Silveira

Sabará  
2024

Morais, Vitoria Rodrigues

M828d

Detecção de falhas em um sistema reator de tanque com agitação contínua utilizando inteligência artificial [manuscrito]. / Vitoria Rodrigues Moraes. - 2024.

56 f., : il.

Orientação: Profa. Dra. Mariella Maia Quadros.

Coorientadora: Prof. Me. Luiz Guilherme Hilel Drummond Silveira.

Trabalho de Conclusão de Curso (Bacharelado em Engenharia de Controle e Automação) – Instituto Federal de Minas Gerais, *Campus* Sabará.

1. Inteligência Artificial. – Monografia. 2. Redes neurais (Computação). – Monografia. 3. Localização de falhas (Engenharia). – Monografia. 4. Controle automático. – Monografia. I. Murofushi, Rodrigo Hiroshi. II. Silveira, Luiz Guilherme Hilel Drummond. III. Instituto Federal de Minas Gerais, *Campus* Sabará. IV. Bacharelado em Engenharia de Controle e Automação. V. Título.

CDU 681.5



**MINISTÉRIO DA EDUCAÇÃO**  
**SECRETARIA DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA**  
**INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE MINAS GERAIS**  
**Campus Sabará**  
**Diretoria de Ensino, Pesquisa e Extensão**  
**Conselho de Área - Controle e Processos Industriais**  
Rodovia MGC 262, Km 10 - Bairro Sobradinho - CEP 34590-390 - Sabará - MG  
- www.ifmg.edu.br

## **ATA DE DEFESA DO TCC**

### **BACHARELADO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO**

Ao dia 09 do mês de outubro do ano de 2024, às 19 horas, sob a presidência de Mariella Maia Quadros, a discente **Vitória Rodrigues Morais** do Curso de Engenharia de Controle e Automação, R.A nº 0049903 do IFMG – Campus Sabará, defendeu o Trabalho de Conclusão de Curso (TCC) intitulado “Detecção de Falhas em um Sistema Reator de Tanque com Agitação Contínua Utilizando Inteligência Artificial” e foi avaliado com a nota final média de 90 pontos, que está condicionada ao cumprimento dos procedimentos pós-defesa do TCC.

Compuseram a Banca Examinadora:

Membro 1 - Prof. Dra. Mariella Maia Quadros - IFMG *Campus Sabará* - Orientadora

Membro 2 - Prof. Dr. Carlos Alexandre Silva - IFMG *Campus Sabará*

Membro 3 - Prof. Dr. Moisés Henrique Ramos Pereira - IFMG *Campus Ribeirão das Neves*

O discente deverá apresentar o trabalho com as devidas modificações em formato pdf, até 31/10/2024 à Coordenação de TCC e fazer o depósito no repositório institucional de TCC do IFMG. O não cumprimento dos procedimentos pós-defesa de TCC até a data estipulada implica no não cumprimento das horas referentes aos componentes curriculares de TCC I e II.

Sabará, 01 de novembro de 2024.



Documento assinado eletronicamente por **Mariella Maia Quadros, Professora EBTT**, em 01/11/2024, às 16:10, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Carlos Alexandre Silva, Professor EBTT**, em 03/11/2024, às 08:09, conforme Decreto nº 10.543, de 13 de novembro de 2020.



Documento assinado eletronicamente por **Moises Henrique Ramos Pereira, Professor**, em 03/04/2025, às 21:02, conforme Decreto nº 10.543, de 13 de novembro de 2020.



A autenticidade do documento pode ser conferida no site <https://sei.ifmg.edu.br/consultadocs> informando o código verificador **2090788** e o código CRC **FEF258C0**.

Dedico este trabalho a minha tia Graça Ferraz por ter sido minha referência nos estudos, e ter plantado em mim o apreço pelo mundo dos estudos e da pesquisa.

## **AGRADECIMENTOS**

Gostaria de agradecer primeiramente aos meus orientadores Mariella Maia e Luiz Silveira por me orientarem durante todo o processo de desenvolvimento e escrita do TCC, aos professores de Engenharia de Controle e Automação do IFMG por, não só ensinarem tanto, mas por estarem sempre disponíveis para ajudar e serem sempre compreensíveis. Também gostaria de agradecer a minha família, amigos e namorado por terem me apoiado durante toda a faculdade.

## RESUMO

A crescente automatização dos processos industriais estimula a busca por sistemas estáveis e confiáveis. No entanto, a ocorrência de falhas em dispositivos industriais é uma realidade, que pode causar danos significativos à segurança, meio ambiente e desempenho do sistema. Este trabalho propõe a aplicação da Rede Neural *Multilayer Perceptron* como método de detecção e diagnóstico de falhas em um sistema de reator de tanque com agitação contínua (CSTR, do inglês *Continuous Stirred Tank Reactor*), um sistema com ampla utilização na indústria. O objetivo é desenvolver um sistema de estimação de falhas para atuar como um módulo de detecção e isolamento de falhas (FDI, do inglês *Fault and Isolation Diagnosis*), capaz de fornecer informações a um sistema de Controle Tolerante a Falhas (FTC, do inglês *Fault Tolerant Control*). Esse sistema visa manter o funcionamento do CSTR de forma segura e com boas métricas de desempenho, mesmo em condições de falha. A pesquisa contribui para a evolução de métodos de detecção e diagnóstico de falhas em sistemas dinâmicos controlados e destaca o papel essencial da Inteligência Artificial na garantia de operações confiáveis e seguras no controle de processos industriais.

**Palavras-chave:** Inteligência Artificial; Redes Neurais; *Multilayer Perceptron*; Controlador Tolerante a Falhas.

## ABSTRACT

The increasing automation of industrial processes stimulates the search for stable and reliable systems. However, the occurrence of failures in industrial devices is a reality, which can cause significant damage to safety, the environment and system performance. This proposes the application of the Multilayer Perceptron Neural Network as a method for detecting and diagnosing faults in a Continuous Stirred Tank Reactor (CSTR) system, a system that is widely used in industry. The aim is to develop a fault estimation system to act as a Fault Detection and Isolation Diagnosis (FDI) module, capable of providing information to a Fault Tolerant Control (FTC) system. This system aims to keep the CSTR operating safely and with good performance metrics, even under fault conditions. A research contributes to the evolution of methods for detecting and diagnosing faults in controlled dynamic controlled systems and highlights the essential role of Artificial Intelligence in guaranteeing reliable and safe operations in industrial process control.

**Keywords:** Artificial Intelligence; Fault Detection; Neural Network; Multilayer Perceptron; Fault Tolerant Control.

## LISTA DE ILUSTRAÇÕES

Figura 1 – Reator Contínuo de Tanque Continuamente Agitado. . . . .	21
Figura 2 – Simulação do Sistema CSTR e suas falhas. . . . .	22
Figura 3 – Neurônio. . . . .	24
Figura 4 – Rede Neural de uma Camada. . . . .	24
Figura 5 – Correlação das Variáveis do Sistema. . . . .	34
Figura 6 – Correlação das Variáveis dentro da Classe de Falha 1, falha de processo. . .	35
Figura 7 – Classe 0, Funcionamento Nominal do Sistema. . . . .	36
Figura 8 – Classe 1 - Falha na Catálise. . . . .	37
Figura 9 – Classe 2 - Falha na Transferência de Calor. . . . .	38
Figura 10 – Classe 3 - Distúrbio na Leitura do Sensor 1. . . . .	38
Figura 11 – Classe 4 - Distúrbio na Leitura do Sensor 2. . . . .	39
Figura 12 – Classe 5 - Distúrbio na Leitura do Sensor 3. . . . .	40
Figura 13 – Classe 6, Distúrbio na Leitura do Sensor 4. . . . .	41
Figura 14 – Classe 7, Distúrbio na Leitura do Sensor 5. . . . .	41
Figura 15 – Classe 8, Distúrbio na Leitura do Sensor 6. . . . .	42
Figura 16 – Classe 9, Distúrbio na Leitura do Sensor 7. . . . .	42
Figura 17 – Acurácia em relação à quantidade de neurônios nas camadas ocultas. . . . .	44
Figura 18 – Matriz Confusão - 4 neurônios na camada oculta. . . . .	45
Figura 19 – Matriz Confusão - 8 neurônios na camada oculta. . . . .	46
Figura 20 – Matriz Confusão - (4,8) neurônios na camada oculta. . . . .	46
Figura 21 – Matriz Confusão - (8,8,8) neurônios na camada oculta. . . . .	47
Figura 22 – Acurácia em relação à quantidade de neurônios e camadas ocultas. . . . .	48
Figura 23 – Matriz Confusão - 4 neurônios na camada oculta. . . . .	48
Figura 24 – Matriz Confusão do Treinamento com (8,8) neurônios na camada oculta. . .	49
Figura 25 – Matriz Confusão do Treinamento com (8,16,8) neurônios na camada oculta.	50
Figura 26 – Matriz Confusão do Treinamento com (10,20,10) neurônios na camada oculta.	50

## LISTA DE ABREVIATURAS E SIGLAS

IFMG	Instituto Federal de Minas Gerais
FTC	do inglês, <i>Fault Tolerant Control</i>
FDI	do inglês, <i>Fault and Isolation Detection</i>
IA	Inteligência Artificial
CSTR	Reator de Tanque com Agitação Contínua
SVM	Máquinas de Vetores de Suporte
RNN	do inglês, <i>Recurrent Neural Network</i>
ANN	do inglês, <i>Artificial Neural Network</i>
MLP	do inglês, <i>Multilayer Perceptron Neural Network</i>

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b>	<b>14</b>
<b>1.1</b>	<b>Objetivos</b>	<b>15</b>
<i>1.1.1</i>	<i>Objetivo geral</i>	<i>15</i>
<i>1.1.2</i>	<i>Objetivos Especificos</i>	<i>15</i>
<b>1.2</b>	<b>Justificativa</b>	<b>15</b>
<b>1.3</b>	<b>Organização do Texto</b>	<b>16</b>
<b>2</b>	<b>REVISÃO BIBLIOGRÁFICA</b>	<b>18</b>
<b>3</b>	<b>FUNDAMENTAÇÃO TEÓRICA</b>	<b>20</b>
<b>3.1</b>	<i>Sistema CSTR</i>	<i>20</i>
<b>3.2</b>	<i>Estudo do Sistema</i>	<i>21</i>
<b>3.3</b>	<i>Rede Neural MLP Classifier</i>	<i>23</i>
<b>4</b>	<b>METODOLOGIA</b>	<b>26</b>
<b>4.1</b>	<b>Análise e Entendimento do Banco de Dados do Sistema CSTR</b>	<b>26</b>
<b>4.2</b>	<b>Pré-Processamento</b>	<b>28</b>
<b>4.3</b>	<b>Treinamento e Teste da Rede Neural</b>	<b>29</b>
<b>4.4</b>	<b>Avaliação de Desempenho</b>	<b>30</b>
<b>5</b>	<b>RESULTADOS</b>	<b>32</b>
<b>5.1</b>	<b>Análise Exploratória do Banco de Dados</b>	<b>32</b>
<i>5.1.1</i>	<i>Análise Quantitativa e Estatística</i>	<i>32</i>
<i>5.1.2</i>	<i>Correlação de Variáveis</i>	<i>34</i>
<i>5.1.3</i>	<i>Análise Comportamental das Variáveis</i>	<i>36</i>
<b>5.2</b>	<b>Treinamento e Testes da MLP</b>	<b>43</b>
<i>5.2.1</i>	<i>Treinamento e Teste no Domínio Base</i>	<i>43</i>
<i>5.2.2</i>	<i>Treinamento e Teste Todos os Domínios</i>	<i>47</i>
<b>5.3</b>	<b>Análise comparativa</b>	<b>51</b>
<b>6</b>	<b>CONCLUSÃO</b>	<b>53</b>
<i>6.0.1</i>	<i>Trabalhos Futuros</i>	<i>53</i>
	<b>REFERÊNCIAS</b>	<b>55</b>

# 1 INTRODUÇÃO

A automatização dos processos industriais tem crescido nas últimas décadas, havendo assim um aumento da complexidade e automação dos dispositivos utilizados, sendo necessário que cada vez mais esses sistemas apresentem estabilidade e confiabilidade (MARTINS *et al.*, 2022).

No entanto, grande parte dos sistemas utilizados nos processos industriais estão suscetíveis à falhas, seja devido a erros no modelo, mau funcionamento de sensores e atuadores, ou perturbações externas (STEFFEN, 2005). Essas falhas são comportamentos não esperados ou desvios que podem acontecer em um sistema (MONTESUMA *et al.*, 2022a). Esses desvios quando não são identificados e tratados da forma correta podem evoluir para uma quebra, ocasionando possíveis danos, como instabilidade no sistema, riscos à segurança, impactos ambientais, entre outros.

Devido aos impactos negativos que as falhas podem gerar em processos industriais, nos últimos anos tem sido notado um crescimento de pesquisas relacionadas às técnicas de detecção e diagnóstico de falhas e às formas de correção dessas falhas. Uma das alternativas para esse problema é a utilização de controladores tolerantes a falhas (FTC, do inglês *Fault Tolerant Control*), que garantem que mesmo na presença de falhas, o sistema permanecerá estável, prevenindo assim que danos mais severos ocorram (QUADROS; LEITE; PALHARES, 2022).

Em técnicas mais avançadas, o controlador tolerante a falhas implementado precisa receber informações sobre as falhas do sistema para que sejam capazes de modificar o laço de controle, sendo essa informação fornecida pelo sistema de detecção e isolamento de falhas (FDI, do inglês *Fault Detection and Isolation*) (QUADROS *et al.*, 2020). Dessa forma, um módulo FDI deve ser adicionado ao sistema de malha fechada, sendo o responsável por garantir o fornecimento de dados precisos sobre as falhas ao controlador projetado.

O módulo FDI é um bloco que funciona de forma independente do controlador, visto que o mesmo não necessita de informações do controlador para estimar as falhas. Portanto, a técnica escolhida para detectar e diagnosticar as falhas deve ser escolhida de acordo com a natureza do problema e os objetivos a serem alcançados, pois cada técnica se adapta melhor a um tipo de sistema e suas características de falha.

A Inteligência Artificial (IA), existente desde 1956 e definida logo após a segunda guerra mundial, é o campo que tenta entender, compreender e construir entidades inteligentes, e sua definição ao longo dos anos se moldou a diversas vertentes de estudos (RUSSEL; NORVIG, 2013). Atualmente é um dos campos mais estudados dentro da ciência e engenharia, e apesar de seu conceito existir desde 1956 ainda é uma área que está em constante expansão.

Como citado acima, processos industriais como um todo vêm se modernizando ao longo do tempo e para este projeto, o sistema escolhido foi um reator contínuo de tanque agitado (CSTR, do inglês *Continuous Stirred Tank Reactor*). O CSTR é um sistema dinâmico amplamente utilizado

no contexto industrial, mais especificamente em indústrias químicas e processos nucleares, além de ter uma grande aplicação em materiais didáticos de controle, sendo utilizado por exemplo, em Ogata (2010), referência bibliográfica amplamente utilizada em disciplinas da área de Controle de Sistemas Dinâmicos.

Sendo assim, este trabalho propõe a aplicação de uma inteligência artificial como o método de classificar e diagnosticar as falhas existentes no CSTR, atuando dessa forma como o módulo FDI do sistema. Portanto, o algoritmo de IA será responsável por classificar, diagnosticar e comunicar as informações de falha para o controlador, permitindo assim que o mesmo seja capaz de realizar as mudanças necessárias no loop de controle, de forma a manter o sistema estável e com bom desempenho.

## 1.1 Objetivos

### 1.1.1 *Objetivo geral*

O objetivo deste trabalho é desenvolver um sistema de estimação de falhas para o reator de tanque continuamente agitado (CSTR) por meio da implementação de redes neurais, para detectar e diagnosticar as falhas do mesmo e fornecer informações precisas para um controlador tolerante a falhas, atuando dessa forma como um bloco FDI.

### 1.1.2 *Objetivos Específicos*

Os objetivos específicos deste trabalho são:

- Analisar a dinâmica do sistema CSTR e como as variáveis atuam no mesmo, fazendo assim um levantamento de como podem ser relacionadas às quais falhas.
- Simular o sistema para obter seus dados de funcionamento dentro de um tempo de amostra determinado, considerando simulações em condições normais de trabalho e em condições de falha.
- Projetar, treinar e testar a rede neural e realizar a análise de acurácia da mesma.
- Analisar o desempenho e os resultados obtidos pelo algoritmo implementado, avaliando os prós e contras de sua aplicação para o contexto de detecção e diagnóstico de falhas para o sistema CSTR.

## 1.2 Justificativa

Com o surgimento da indústria 4.0, diversos fenômenos aconteceram dentro do contexto industrial, como a modernização e automação de linhas de produção, a aplicação de sistemas de controle cada vez mais robustos, bem como o armazenamento em grande escala de dados de

máquinas. Estima-se, por exemplo, que os dados armazenados entre 2015 e 2018 tenham sido o dobro dos armazenados entre 1995 e 2015 (GONÇALVES, 2022).

Devido a esse aumento no armazenamento de históricos de funcionamento das linhas de produção, iniciou-se um processo de aproveitamento desses históricos para aumentar o entendimento de diversos dispositivos, assim como enfatizou as pesquisas de técnicas de *Machine Learning* e Inteligência Artificial (JÚNIOR, 2022). Isso porque essas técnicas consistem basicamente em treinar algoritmos para reconhecerem os padrões de funcionamento de máquinas e assim serem capazes de prever acontecimento futuros, e também classificar e diagnosticar desvios nesse funcionamento (RUSSEL; NORVIG, 2013).

Como citado anteriormente, os sistemas no geral estão suscetíveis a falhas durante sua operação. No entanto, os reatores por se tratarem de sistemas químicos, podem causar verdadeiras catástrofes, como por exemplo o desastre de Chernobyl, ocorrido nos dias 25 e 26 de abril de 1986, em que um reator nuclear teve uma sucessão de falhas durante um teste de segurança, que o levou ao um mau funcionamento e eventualmente a uma catástrofe que liberou cerca de 200 toneladas de material radioativo na atmosfera, tornando a cidade em que o mesmo era localizado inabitável até os dias atuais (DROZDOVITCH, 2021).

Sendo assim, os sistemas de previsão de falhas possuem uma grande relevância no ambiente industrial, pois além de serem capazes de aumentar a eficiência de sistemas, evitar mau funcionamento das máquinas, também têm um papel importante na garantia de segurança do sistema. E por conta da evolução na forma de adquirir e armazenar dados ocasionados pela indústria 4.0, a inteligência artificial se tornou um campo com uma ampla gama de possibilidades.

Além disso, a engenharia de controle e automação ocupa um papel importante no cenário de previsões de falhas em sistemas dinâmicos, visto que integra conhecimentos como automação, controle de sistemas dinâmicos, programação, circuitos elétricos, conceitos de física ligados a fluidos e trocas de calor e à inteligência artificial. Sendo assim, o engenheiro de controle e automação possui diversos conhecimentos relevantes para atuar de forma a minimizar e prevenir falhas em diversos sistemas, sendo assim capaz de garantir a confiabilidade dos sistemas presentes em diversas indústrias.

### 1.3 Organização do Texto

Este trabalho está organizado da seguinte forma:

- (a) Introdução, em que o tema e a aplicação são brevemente explicados.
- (b) Revisão Bibliográfica, capítulo em que trabalhos similares envolvendo inteligência artificial são citados e exemplificados.

- 
- (c) Fundamentação Teórica, neste capítulo são apresentados os conhecimentos necessários para um bom entendimento do decorrer do trabalho.
  - (d) Metodologia, em que são apresentados os métodos que serão utilizados para desenvolver o trabalho.
  - (e) Resultado, onde serão apresentados os resultados obtidos durante o desenvolvimento do trabalho.
  - (f) Conclusão, serão apresentadas as conclusões referentes a metodologia desenvolvida e os resultados obtidos.
  - (g) Referências, em que está contido todos os artigos e livros que baseiam este trabalho.

## 2 REVISÃO BIBLIOGRÁFICA

A inteligência artificial é um campo que vem sendo amplamente estudado desde os meados do século XXI, por pesquisadores de diversas áreas, como matemática, robótica, neurociência e engenharia computacional (GONTIJO *et al.*, 2020). Por conta da complexidade dos processos químicos, a aplicação de técnicas de inteligência artificial para reduzir custos e aumentar a segurança de plantas químicas tem sido uma área de estudo muito ativa (KANG, 2020).

Dentre trabalhos que tem o objetivo de detectar e diagnosticar falhas, diversos utilizam a rede neural como método, como GONÇALVES (2022), Andrade (2021), Juvêncio *et al.* (2023), Kang (2020) e Schnitman (1998), assim como trabalhos que fazem um comparativo entre diversas aplicações como árvores de decisão (VIEIRA, 2021), lógica Fuzzy (JÚNIOR, 2022), Redes Neurais (JÚNIOR, 2022), Máquinas de Vetores de Suporte (TELES; FREITAS; RODRIGUES, 2019), entre outros, assim como trabalhos que utilizam CVA (do inglês, *Canonical Variate Dissimilarity Analysis*) (PILARIO; CAO, 2018).

No entanto, para esta revisão bibliográfica será dado enfoque em três trabalhos que podem ser relacionados ao sistema e objetivos que serão utilizados no decorrer deste trabalho, conforme apresentado a seguir.

Em Kang (2020), o autor faz um comparativo do desempenho de dois tipos de Redes Neurais, RNNs (do inglês, *Recursive Neural Networks*) e ANNs (do inglês, *Artificial Neural Networks*), na detecção de falhas em processos químicos complexos. No trabalho foi avaliado um total de 21 falhas e os métodos treinados com dois bancos de dados diferentes, um com 5 amostras e outro com 40. Como resultado, foi obtido que a RNN teve um índice de acerto de 84,9% se treinada com somente 5 amostras e 92,5% se treinada com os 40 amostras, já a ANN obteve 75,5% e 87,3%, respectivamente. Portanto, os resultados obtidos indicam que a rede do tipo RNN teve uma quantidade maior de acertos quando comparada com a rede ANN, indicando assim uma maior capacidade de identificar falhas ocorridas no sistema em que os métodos foram aplicados.

Já em Amin e Hasan (2019), foi aplicada a metodologia de lógica Fuzzy como módulo FDI para um controlador tolerante a falhas, em um motor de combustão à gasolina. Como resultado de suas simulações, foi obtido um comportamento estável do motor, mantendo os indicadores de desempenho todos dentro do especificado pelo problema, em que o sistema não apresentou nenhuma degradação mesmo durante a simulação de diversas falhas e distúrbios.

Apesar de ser comum a utilização ou de RNNs ou de lógica Fuzzy, também existem trabalhos que utilizam um método neural-Fuzzy, como em Chhipa *et al.* (2021), em que o autor aplicou esse método para atuar como o FDI de um sistema. Nesse caso, o objetivo era manter a estabilidade de um sistema de geradores movidos a turbinas eólicas, em que o mesmo se mantém trabalhando em um ponto de máxima velocidade e eficiência. Como conclusão de suas simulações, o controlador obteve uma resposta rápida para mudanças de velocidades e conseguiu

manter o sistema estável durante todo seu funcionamento.

### 3 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, será abordada a fundamentação teórica necessária para compreender o contexto do trabalho e as técnicas utilizadas na aplicação de inteligência artificial para a previsão de falhas em sistemas dinâmicos, com foco no reator revestido do tipo CSTR e a classificação de falhas em problemas com múltiplos domínios.

#### 3.1 Sistema CSTR

Sistema é a combinação de vários componentes que trabalham de forma a atingir um objetivo em comum, podendo ser eles componentes físicos ou não. O sistema CSTR (do inglês, *Continuous Stirred-Tank Reactor*) é um exemplo de sistema no qual ocorre uma reação exotérmica de primeira ordem de A, para B, em que A e B são dois reagentes químicos. O comportamento desse sistema pode ser modelado considerando três equações diferenciais ordinárias (ODEs) que descrevem as variáveis C, T e  $T_c$  (PILARIO; CAO, 2018), como descritas nas equações abaixo:

$$\Sigma_P = \begin{cases} \dot{C} = \frac{Q}{V}(C_i - C) - akC^N + v_1, \\ \dot{T} = \frac{Q}{V}(T_i - T) - a \left( \frac{\delta H_r k C^N}{\rho C_p V} \right) - b \frac{UA}{\rho C_p V} (T - T_c) + v_2, \\ \dot{T}_c = \frac{Q_c}{V_c} (T_{ci} - T_c) - b \frac{U^p A}{\rho_c C_{pc} V_c} (T - T_c) + v_3. \end{cases} \quad (3.1)$$

Na Figura 1 é mostrado o diagrama do CSTR e suas variáveis.

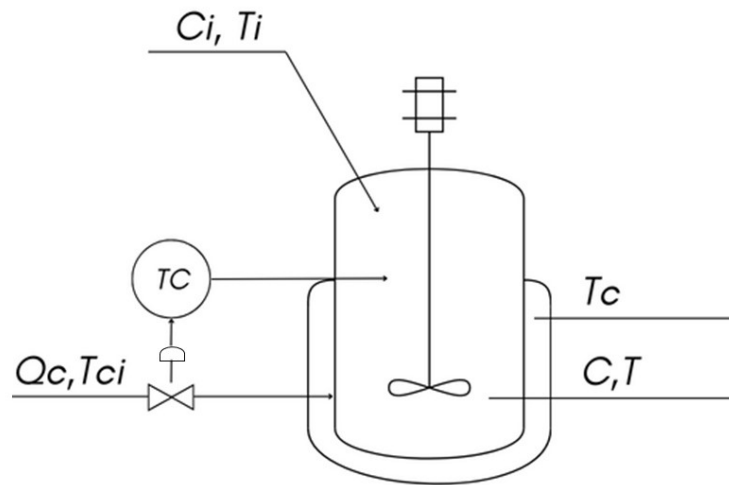
As variáveis apresentadas na Figura 1 representam o conjunto de entradas e saídas do sistema, conforme especificado na Tabela 1.

Tabela 1 – Variáveis de entrada e saída do CSTR.

Variável	Descrição	Unidade de Medida
$C_i$	Concentração de A	mol/L
$T_i$	Temperatura de entrada de A	K
$Q_{ci}$	Fluxo do líquido refrigerante	L/min
$T_{ci}$	Temperatura de entrada do líquido refrigerante	K
C	Concentração de B na saída	mol/L
T	Temperatura do tanque	K
$T_c$	Temperatura de saída de B	K

Sendo assim, o CSTR é um sistema cuja inteligência artificial pode atuar na classificação de falhas relacionadas às suas 7 variáveis, identificando o tipo de falha a partir da combinação amostral das mesmas.

Figura 1 – Reator Contínuo de Tanque Continuamente Agitado.



Fonte: Adaptado de Montesuma *et al.* (2022b).

### 3.2 Estudo do Sistema

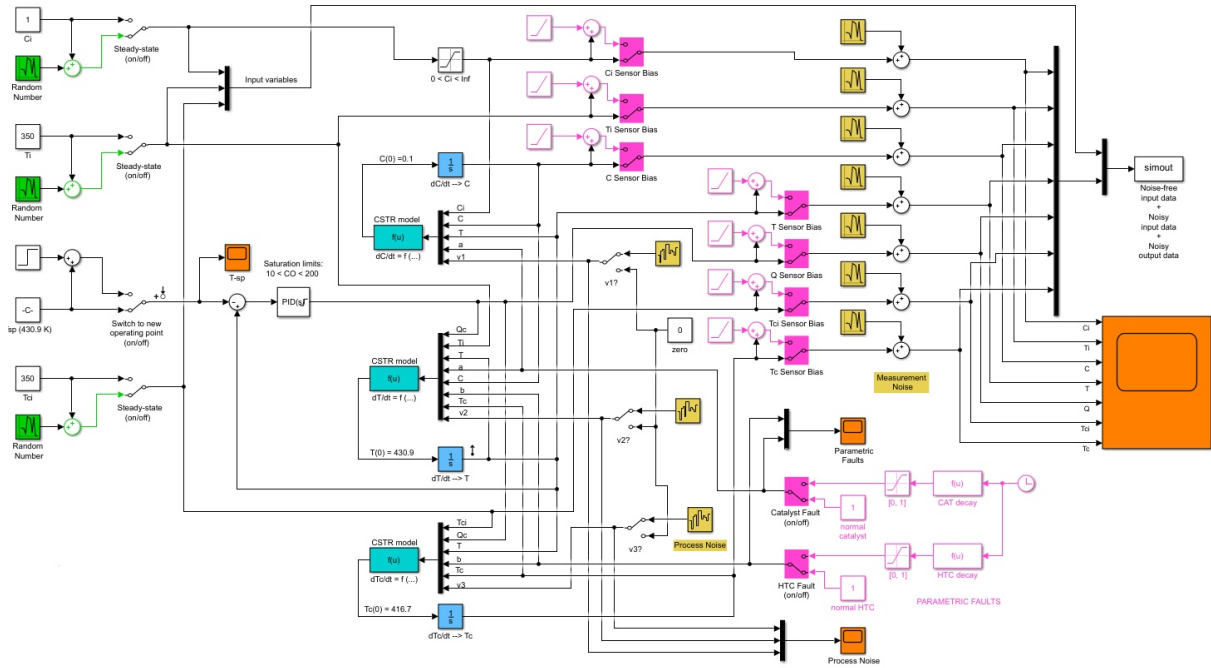
Por ser de grande utilização na literatura, simulações do sistema CSTR estão disponíveis em diversos fóruns de ciência de dados na internet. Dessa forma, neste trabalho foi escolhido um banco de dados e diagrama de simulação utilizados em três trabalhos: (PILARIO; CAO, 2018), (MONTESUMA, 2021) e (MONTESUMA *et al.*, 2022b). Os arquivos estão disponíveis em Pilario (2023) e Kaggle (2024).

Na Figura 2 é apresentado o diagrama da simulação do sistema CSTR.

Esta simulação está subdividida em partes, em que cada uma delas é representada por uma cor, como exemplificado na Tabela 2. A cada simulação executada, são geradas 200 amostras de dados, com um tempo de amostragem de 1 minuto. Ou seja, a cada simulação teremos um total de 200 amostras de cada uma das variáveis monitoradas. A adição de falhas e distúrbios no sistema é feita de forma manual ao selecionar as chaves, que podem ser de distúrbios ou seletoras de falha, ou seja, a geração de dados de falha pode ser feita de acordo com a necessidade do usuário.

Portanto, com base nas equações e simulação, pode-se concluir que as variáveis de saída são  $\hat{C}$ ,  $\hat{T}$  e  $\hat{T}_c$ , enquanto que as variáveis de entrada são  $C_i$ ,  $T_i$ ,  $T_{ci}$  e  $\hat{Q}_c$ . Sendo assim, todos os outros componentes das equações são constantes no funcionamento do sistema, sendo setadas como configurações de funcionamento, que podem, ou não, serem alteradas para geração de diferentes domínios de simulação, como veremos mais a frente. Com base nisso foi feito um

Figura 2 – Simulação do Sistema CSTR e suas falhas.



Retirado de: (PILARIO; CAO, 2018)

Tabela 2 – Identificação dos Componentes da Simulação.

Legenda	Descrição	Cor
Seletores de Falha	São componentes responsáveis por falhas no sistema	Rosa
Modelo do Sistema	Responsável pela representação do Reator	Azul Claro
Integradores das ODEs	São os integradores do sistema	Azul Escuro
Aditores de Ruído	São as entradas que des ruídos do sistema	Amarelo
Distúrbios	São as entradas de distúrbios no sistema	Verde
Escopo	Tempo de simulação	Laranja

levantamento relacionando as variáveis e seus tipos, mostrado na Tabela 3.

Tabela 3 – Variáveis do Sistema CSTR.

Parâmetros	Descrição	SI	Tipo
Q	Fluxo de Entrada	L/min	Parâmetros
V	Volume do Tanque	L	
V <sub>c</sub>	Volume do Tanque de Refrigeração	L	
δ. Hr	Temperatura da Reação	cal/mol	
UA	Coefficiente de Transferência de Calor	K.cal/mol	
K <sub>0</sub>	Fator pré exponencial de K	min <sup>-1</sup>	
E/R	Energia de Ativação sobre Constante de Gás	K	
ρ, ρ <sub>c</sub>	Densidade dos fluidos	g/L	
C <sub>p</sub> , C <sub>pc</sub>	Capacidade calorífica dos fluidos	K.cal/g	
N	Ordem da Reação	-	
C <sub>i</sub>	Concentração de Entrada	mol/L	
T <sub>i</sub>	Temperatura de Entrada	K	
T <sub>ci</sub>	Temperatura de Entrada do Líquido de Arrefecimento	K	
Q <sub>c</sub>	Fluxo/Caudal do Líquido de Arrefecimento	L/min	
C	Concentração de Saída	mol/L	Saídas
T	Temperatura de Saída	K	
T <sub>c</sub>	Temperatura de Saída do Líquido de Arrefecimento	K	

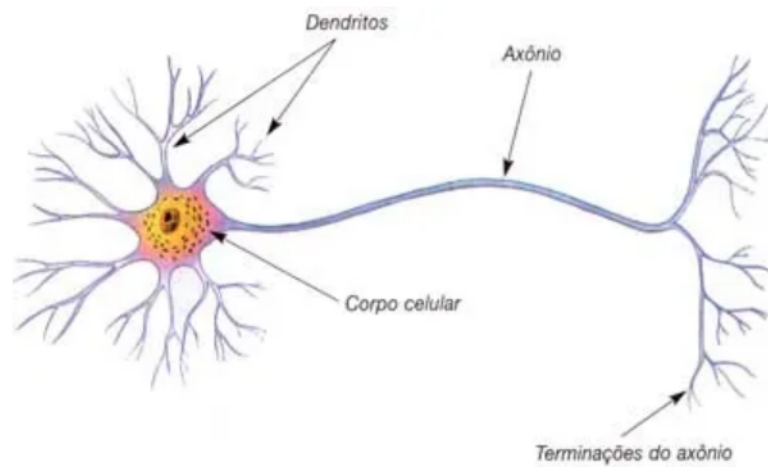
### 3.3 Rede Neural MLP Classifier

Ainda no contexto da indústria 4.0, uma das características apresentadas nesse processo foi o início do armazenamento dos dados de funcionamento das máquinas. Sendo assim, a automação dos processos industriais trouxe também essa característica de armazenar as informações do comportamento dos dispositivos industriais de um determinado tempo (JÚNIOR, 2022). Esse movimento se aliou ao crescimento de estudos no âmbito da inteligência artificial, pois os métodos existentes hoje em dia, em sua maioria, requerem o treinamento do algoritmo a partir do histórico de funcionamento do sistema que desejamos monitorar (GONÇALVES, 2022).

Nesses estudos, temos o avanço de diversas técnicas conhecidas de algoritmos como inteligência artificial, máquinas de vetores de suporte, *Naive Bayes*, árvores de decisão e florestas aleatórias e as chamadas redes neurais artificiais. No entanto, para o objetivo deste trabalho, neste tópico, será explorada a técnica das Redes Neurais Artificiais (RNAs), que tem sido amplamente aplicada na detecção e diagnóstico de falhas em sistemas dinâmicos complexos, como o reator revestido do tipo CSTR. A rede neural têm o potencial de melhorar significativamente a capacidade de identificar falhas em tempo real e auxiliar no processo de tomada de decisões (ANDRADE, 2021).

As RNAs são algoritmos inspirados no funcionamento do cérebro humano, se utilizando do conhecimento acerca da transmissão de informação nos neurônios, onde os dendritos recebem a informação, o corpo celular executa um cálculo e transfere essa informação através de impulsos nervosos pelo axônio Juvêncio *et al.* (2023).

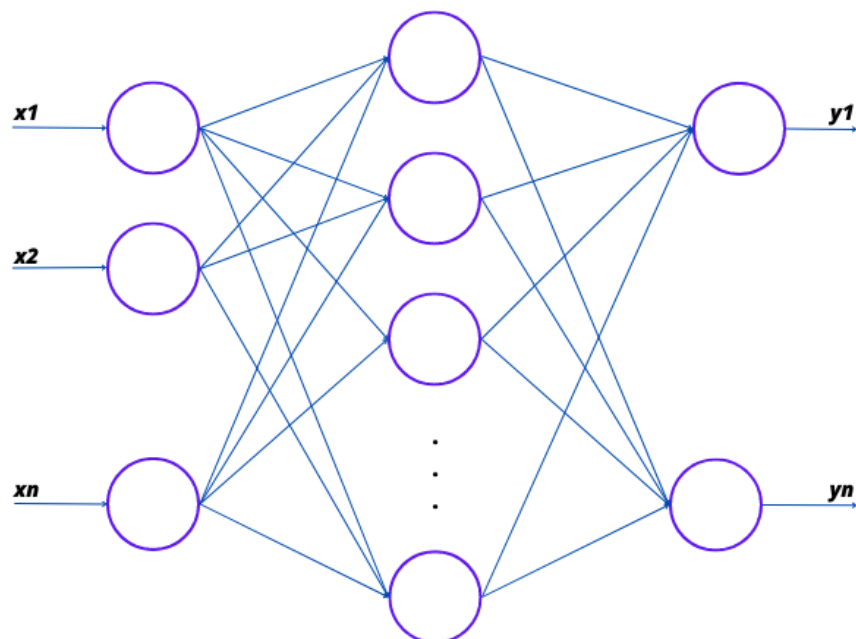
Figura 3 – Neurônio.



Fonte: Adaptado de GONÇALVES (2022)

Baseando-se nesse funcionamento, a rede neural artificial é a combinação de diversos neurônios recebendo, calculando e transmitindo informações, assim como em uma rede neural natural. Portanto, dentro da estrutura das redes neurais também estão presentes os neurônios e os axônios, que podem compor diversas camadas, de acordo com a necessidade do problema, como simplificado na Figura 4.

Figura 4 – Rede Neural de uma Camada.



Fonte: Adaptado de Juvêncio *et al.* (2023)

Por conta dessa formação de rede, com diversas camadas de redistribuição de informação, as RNAs tem a capacidade de aprender relações complexas de entradas e saídas a partir do

treinamento com dados históricos de um determinado processo, podendo ser aplicadas na detecção e diagnóstico de falhas, capturando padrões sutis nos dados que indicam comportamentos anômalos (JUVÊNCIO *et al.*, 2023).

Dentro do contexto das redes neurais artificiais, existem 3 principais estruturas utilizadas em diversas bibliografias, a seguir será exemplificado suas características e funcionamento.

1. Perceptrons de múltiplas camadas (MLPs, do inglês Multilayer Perceptron Neural Network): As redes neurais multicamadas perceptron são redes que possuem diversas camadas ocultas, e podem ter diversos neurônios na camada de saída. Comumente é utilizada a atualização do tipo *backpropagation*, em que a direção de atualização dos pesos ocorre no sentido da camada de saída para a camada de entrada, (PARK; LEK, 2016).
2. Redes Neurais Convolucionais (CNNs, do inglês Convolutional Neural Networks): As redes neurais convolucionais são geralmente utilizadas na resolução de problemas envolvendo processamento e classificação de imagens, como sistemas de visão computacional e sistemas de reconhecimento facial, (FERREIRA, 2020).
3. Redes Neurais Recorrentes (RNNs, do inglês Recurrent Neural Networks): As redes neurais recorrentes são redes que utilizam dados sequenciais, por exemplo, uma série temporal. Portanto, esse tipo de rede é mais comumente utilizado para solucionar problemas temporais, que necessitam que o sistema tenha uma memória temporal sobre o problema. A RNN se diferencia das demais exatamente por ter memória, onde em sua estrutura são recebidos dados passados para influenciar na tomada de decisão presente, enquanto as demais assumem que os dados de passado e presente são independentes, (BENINI, 2008).

## 4 METODOLOGIA

No Capítulo 3 foi apresentado o sistema CSTR e o diagrama de simulação utilizado pelo autor Montesuma *et al.* (2022b) para geração do banco de dados utilizado neste trabalho, e as variáveis do sistema. Neste capítulo serão apresentados os métodos utilizados em cada etapa do trabalho. Assim, o capítulo será dividido em:

- Análise e Entendimento do Sistema
- Pré-Processamento
- Treinamento da Rede
- Avaliação de desempenho

### 4.1 Análise e Entendimento do Banco de Dados do Sistema CSTR

O banco de dados consiste em 7 variáveis observadas em 2860 simulações, onde as 200 amostras de cada variável são dispostas de forma subsequente, produzindo dessa forma um banco de dados de 2860 linhas e 1404 colunas, e ao final são inseridas em cada simulação informações de classe de falha, domínio, ruído e ordem da reação.

Neste estudo de caso são abordados três tipos de falhas: falhas de processo, falhas em sensores e falhas de distúrbios, sendo possível diferenciá-las no diagrama de simulação utilizando as informações contidas na Tabela 2. Além disso, são ao todo 13 classes, em que a classe 0 é o funcionamento nominal do sistema, e o restante das 12 classes são classes de falha. A Tabela 4.1 apresenta cada classe de falha de cada um dos 3 tipos de falha, seu modelo e sua descrição.

Classe	Tipo de Falha	Modelo	Delta	Descrição
1	Processo	$a = a_0 \exp(-\delta t)$	0.004	Falha na Catálise
2		$b = b_0 \exp(-\delta t)$	0.005	Queda na Transferência de Calor
3	Sensores	$\tilde{C}_i = C_i + \delta t$	0.005	Distúrbios na Leitura do sensor 1
4		$\tilde{T}_i = T_i + \delta t$	0.1	Distúrbios na Leitura do sensor 2
5		$\tilde{T}_{ci} = T_{ci} + \delta t$	0.1	Distúrbios na Leitura do sensor 3
6		$\tilde{Q}_c = Q_c + \delta t$	-0.2	Distúrbios na Leitura do sensor 7
7		$\tilde{C} = C + \delta t$	0.005	Distúrbios na Leitura do sensor 4
8		$\tilde{T} = T + \delta t$	0.1	Distúrbios na Leitura do sensor 5
9		$\tilde{T}_c = T_c + \delta t$	0.1	Distúrbios na Leitura do sensor 6
10	Distúrbios	$\Delta C_i + N(0, \delta)$	0.005	Distúrbios em $C_i$
11		$\Delta T_i + N(0, \delta)$	5	Distúrbios em $T_i$
12		$\Delta T_{ci} + N(0, \delta)$	5	Distúrbios em $T_{ci}$

A simulação realizada pelo autor Montesuma *et al.* (2022b) considera a inserção de falha nas classes 1 a 9 a partir da amostra 0, enquanto para as classes 10 a 12 a inserção de falha acontece a cada 60 amostras de simulação. Portanto para as simulações de falha de classes 1 a 9 as

200 amostras das variáveis presentes no sistema serão consideradas amostras de falha, enquanto para as falhas de classes 10 a 12 somente as amostras 60, 120 e 180 serão consideradas de falha. Essa diferença de inserção de falhas para as classes de falha de distúrbio no sistema faz com que a quantidade de amostras de falhas das classes 10 a 12 seja consideravelmente menor que todas as demais classes, por conta disso neste trabalho as classes 10 a 12 serão desconsideradas como falhas, assim como em (PILARIO; CAO, 2018).

Além das 19 classes de falha consideradas, neste estudo de caso estão presentes ao todo 6 domínios de simulação. Cada domínio consiste em uma combinação de parâmetros configurados no sistema, que por estarem presentes nas equações diferenciais ordinárias que regem o sistema, apresentadas no Capítulo 3, alteram o funcionamento do mesmo.

As variáveis alteradas pelos autores Pilario e Cao (2018), Montesuma *et al.* (2022b) e Pilario (2023) na produção de domínios diferentes de simulação estão apresentadas na Tabela 4.1.

Tabela 4 – Variáveis Modificadas na Produção de Domínios.

Parâmetros	Descrição	SI
V	Volume do Tanque	L
V <sub>c</sub>	Volume do Tanque de refrigeração	L
δ. Hr	Temperatura da Reação	cal/mil
UA	Coefficiente de Transferência de calor	K.cal/mol
K <sub>0</sub>	Fator Pré Exponencial de K	min <sup>-1</sup>
E/R	Energia de Ativação sobre constante de gás	K

Dessa forma, a complexidade apresentada nas simulações do sistema se deve não só à grande quantidade de classes de falhas diferentes, mas também ao fato de que as falhas são geradas dentro de domínios diferentes de simulação, apresentando, dessa forma, diversos comportamentos das variáveis presentes no sistema, já apresentadas na Tabela 1.

Portanto, devido à alta variabilidade de classes e domínios, faz-se necessário entender a distribuição das simulações entre classes e domínios, qual o comportamento das variáveis diante das falhas, a escala das variáveis, além da identificação de possíveis problemas de formatação e valores discrepantes no banco de dados.

Para isso, será realizada uma análise exploratória do banco de dados para estudo da distribuição das simulações e amostras dentro do banco, variáveis estatísticas como mediana, desvio padrão, mínimos e máximos de cada variável. Para a realização desse estudo, será feita uma manipulação na estrutura original do banco, onde o mesmo será condensado, pois inicialmente o objetivo da análise será visualizar as variações presentes entre uma simulação e outra e, portanto, será analisada uma média das variáveis por simulação. Dessa forma, o banco de dados anteriormente com 2860 linhas e 1404 colunas passa a ter 2860 linhas e 11 colunas.

Além disso, após o entendimento distributivo e estatístico das simulações contidas no

banco de dados, serão visualizados gráficos do comportamento das variáveis em cada classe presente no banco, de funcionamento nominal e de falha. Nesta visualização o objetivo é entender, de forma generalizada, padrões e tendências existentes nas variáveis.

Para realizar essas manipulações e análises serão utilizadas as seguintes ferramentas:

- Python: linguagem de programação com ampla quantidade de ferramentas e bibliotecas para análise de dados;
- Pandas: biblioteca para manipulação de bancos de dados;
- Matplotlib: biblioteca para plotagem e visualização de gráficos;
- Visual Studio: IDE interativa para programação em diversas linguagens.

Ao final da etapa de análise exploratória serão levantadas as modificações e pré-processamentos necessários para que os dados sejam formatados e inseridos no treinamento da Rede Neural.

## 4.2 Pré-Processamento

Nesta etapa serão realizadas as modificações necessárias para garantir a formatação adequada dos dados antes de iniciar o treinamento da Rede Neural. Primeiramente, o objetivo do trabalho é treinar uma Rede Neural para identificação de falhas minuto a minuto de simulação, de forma que a entrada da rede seja uma amostra de cada variável do sistema. Para isso, é necessário transformar o banco de 2860 linhas e 1404 colunas em um banco de 572000 linhas e 11 colunas. Dessa forma, cada linha passa a ter uma amostra de cada variável, mais as informações de classe, domínio e ruído ao qual a amostra pertence.

Além disso, devido à sensibilidade da rede e à grande diferença de escala entre variáveis, como será mostrado no Capítulo 5, em análise exploratória dos dados, é necessário utilizar métodos de normalização, garantindo assim que todas as variáveis estejam na mesma escala de valores.

Para realizar as modificações necessárias foram utilizadas as seguintes ferramentas:

- Python: linguagem de programação com ampla quantidade de ferramentas para análise de dados.
- Pandas: biblioteca para manipulação de bancos de dados.
- Skicit Learn: biblioteca para escalonamento e treinamento de redes neurais. Dentro desta biblioteca foi utilizada a função `StandardScaler`.
- Visual Studio: IDE interativa para programação em diversas linguagens.

### 4.3 Treinamento e Teste da Rede Neural

Após a análise e pré-processamento dos dados, inicia-se o processo de treinamento da Rede Neural. Dentre os diversos tipos de redes neurais citados no Capítulo 3, o Multilayer Perceptron (MLP) foi escolhido para este trabalho devido à sua ampla aplicabilidade em problemas de classificação, incluindo a classificação multivariada de falhas.

O desempenho da MLP é altamente sensível à configuração de seus hiperparâmetros, e para identificar a melhor combinação destes, será realizada uma série de experimentos, variando o número de neurônios e camadas ocultas. Outros hiperparâmetros, como a função de ativação, otimizador e taxa de aprendizado, foram mantidos constantes em todos os experimentos para isolar o impacto da variação na arquitetura da rede. Os parâmetros escolhidos e mantidos em todos os experimentos estão demonstrados abaixo:

- **Subdivisão de Treinamento e Teste:** Para realização do treinamento e teste da MLP foi executada uma subdivisão dos dados disponíveis no banco de dados, sendo 70% do banco disponibilizado para treinamento e 30% para teste.
- **Função de Ativação (ReLU):** Optou-se pela função de ativação ReLU (do inglês, *Rectified Linear Unit*) devido à sua eficiência em redes neurais profundas. A ReLU introduz não-linearidade na rede, permitindo que ela aprenda representações mais complexas dos dados (LEARN, 2024). Além disso, a ReLU contribui para acelerar o treinamento da rede, evitando o problema da anulação de neurônios, conhecido como *gradient vanish*.
- **Otimizador (Adam):** O algoritmo Adam foi escolhido como otimizador devido à sua adaptabilidade e eficiência. O Adam combina as vantagens dos algoritmos Adagrad e RMSprop, adaptando a taxa de aprendizado para cada parâmetro da rede (LEARN, 2024). Essa adaptação permite que o Adam convirja mais rapidamente para um mínimo local da função de custo, especialmente em problemas com grandes conjuntos de dados e muitos parâmetros.
- **Máxima Iteração (200):** A máxima iteração foi definida em 200, representando o número máximo de vezes que o algoritmo de otimização irá percorrer o conjunto de treinamento (LEARN, 2024). Um número maior de iterações pode levar a um modelo mais preciso, mas também aumenta o tempo de treinamento.
- **Número de Iterações sem Mudança (50):** O critério de parada foi definido em 50 iterações sem mudança nos pesos. Essa estratégia visa evitar o *overfitting*, ou seja, o ajuste excessivo do modelo aos dados de treinamento. Ao interromper o treinamento quando os pesos deixam de ser atualizados significativamente, evita-se que o modelo memorize o ruído presente nos dados e generalize melhor para novos exemplos (LEARN, 2024).

- Taxa de Aprendizado Inicial: A taxa de aprendizado inicial define o tamanho dos passos que o algoritmo de otimização dará durante o treinamento (LEARN, 2024). Uma taxa de aprendizado muito alta pode levar a uma oscilação no processo de otimização, enquanto uma taxa muito baixa pode levar a um treinamento muito lento. A escolha da taxa de aprendizado inicial é crucial para o desempenho da Rede Neural e geralmente requer ajuste fino.

As seguintes ferramentas foram utilizadas no treinamento e teste da MLP:

- Python: linguagem de programação com ampla quantidade de ferramentas e bibliotecas para análise de dados.
- Scikit Learn: biblioteca com diversas funções, entre elas *train\_test\_split*, para subdivisão do banco de dados em treino e teste e *MLPClassifier*, para configuração, treinamento e testes da MLP.
- Visual Studio: IDE interativa para programação em diversas linguagens.

#### 4.4 Avaliação de Desempenho

Após o treinamento da Rede Neural, é necessário realizar a avaliação de seu desempenho. A implementação e análise de métricas permite não apenas validar os resultados obtidos, como também comparar o desempenho de diferentes configurações da rede, identificando dessa forma pontos de ajustes.

As métricas mais comuns utilizadas para avaliar o desempenho de modelos de classificação, como a MLP adotada neste trabalho, são acurácia (JUVÊNCIO *et al.*, 2023) e matriz de confusão (JÚNIOR, 2022). Ambas as métricas fornecem informações sobre a capacidade do modelo de classificar corretamente novas amostras, ambas estão explicadas abaixo:

- Acurácia: a acurácia representa a proporção de classificações corretas em relação ao total de exemplo, apresentando dessa forma a porcentagem de acertos da MLP.
- Matriz de Confusão: a matriz de confusão é uma Tabela que cruza as classes reais com as classes previstas pelo modelo. A mesma permite visualizar a distribuição dos erros cometidos pela MLP, identificando as confusões mais frequentes, revelando possíveis dificuldades na diferenciação entre duas classes.

Dessa forma, neste trabalho ambas as métricas foram escolhidas como método de avaliação, em que acurácia fornece uma visão geral de desempenho e a matriz de confusão apresenta uma análise mais detalhada dos erros cometidos pelo modelo. Em que para melhor visualização

e comparação do desempenho em diferentes treinamentos foram gerados gráficos de acurácia por treinamento, e mapas de calor das matrizes de confusão por treinamento.

Para aplicação e comparação das métricas, foram utilizadas as seguintes ferramentas:

- Python: linguagem de programação com ampla quantidade de ferramentas e bibliotecas para análise de dados.
- Pandas: biblioteca para manipulação de bancos de dados.
- *SkicitLearn.metrics*: biblioteca utilizada para calcular as métricas escolhidas, utilizando-se as funções *accuracy\_score*, *confusion\_matrix*.
- Seaborn: biblioteca para geração de gráficos do tipo mapa de calor.
- *Visual Studio*: IDE interativa para programação em diversas linguagens.

## 5 RESULTADOS

Neste capítulo, serão apresentados os resultados obtidos na análise exploratória dos dados, e posteriormente, no desenvolvimento da rede neural artificial para a classificação de falhas.

### 5.1 Análise Exploratória do Banco de Dados

A análise exploratória do banco de dados tem como objetivo principal compreender o comportamento das variáveis presentes no sistema. A partir da análise estatística das amostras nominais e de falha, é possível entender a distribuição dos dados simulados dentro de cada classe e domínio da simulação, facilitando dessa forma a visualização de padrões e também de possíveis problemas nas amostras presentes no banco de dados. Além disso, a análise inicial dos dados ajuda a determinar os melhores métodos de pré-processamento a serem adotados para o treinamento da rede.

#### 5.1.1 Análise Quantitativa e Estatística

Como exemplificado no Capítulo 3, o banco de dados utilizado neste trabalho consiste em 2860 linhas, cada linha contendo 200 amostras de cada variável do sistema, onde, ao final, mais 4 colunas são adicionadas, informando: classe de falha, domínio da simulação, ruído e ordem da reação química, totalizando assim 1404 colunas. Como evidenciado no Capítulo 3, o ruído e a ordem da reação química são as variáveis utilizadas para gerar as classes e domínios de simulação. Por conta disso, entende-se que suas variações já estão contabilizadas dentro das informações de classe e domínio. No entanto, para visualizar melhor a distribuição dos dados como um todo, essas variáveis serão consideradas na etapa de análise. Portanto, inicialmente, foi necessário entender qual a quantidade de simulações estão presentes em cada classe de falha e domínio, onde foi avaliada a distribuição das simulações entre essas informações, obtendo assim o resultado apresentado na Tabela 5.

Tabela 5 – Tabela de Distribuição dos Dados.

Classe de Falha	0	1	2	3	4	5	6	7	8	9	10	11	12
Quantidade de Simulações por Classe de Falha	220	220	220	220	220	220	220	220	220	220	220	220	220
Domínio	0	1	2	3	4	5	6						
Quantidade de Simulações por Domínio	1300	260	260	260	260	260	260						
Ruído	0	0.1	0.15	0.2									
Quantidade de Simulações por Ruído	1300	260	1040	260									
Ordem da Reação	0.5	1	15	2									
Quantidade de Simulações por Ordem de Reação	260	2080	260	260									

Observa-se que a distribuição das simulações ocorre de forma homogênea entre classes. No entanto essa distribuição ocorre de forma heterogênea para o domínio, ruído e ordem da

reação, apresentando uma alta concentração de simulações em um único cenário de simulação. Essa distribuição desbalanceada dos dados pode ocasionar problemas de treinamento da rede, pois ao ser treinada com uma quantidade desigual de amostras de cada falha, a rede tende a ter um aprendizado também desbalanceado, apresentando uma menor acurácia na classificação de falhas que receberam menores amostras de treinamento. Por conta disso foi necessário entender como é a distribuição dos dados de forma mais aprofundada, visualizando como a quantidade de amostras em uma única classe se distribui entre domínio, ruído e ordem de reação. Para esse estudo mais aprofundado, foi feita a análise de distribuição utilizando-se a classe de falha um, em que o resultado está apresentado na Tabela 6.

Tabela 6 – Tabela de Distribuição das Falhas.

Domínio	0	1	2	3	4	5	6
Quantidade de Simulações	100	20	20	20	20	20	20
Ruído	0	0.1	0.15	0.2			
Quantidade de Simulações	100	20	80	20			
Ordem da Reação	0.5	1	15	2			
Quantidade de Simulações	20	160	20	20			

Pode-se observar na Tabela 6 que a distribuição heterogênea evidenciada na análise mais superficial do banco se manteve, mostrando que grande parte das simulações foram executadas dentro do cenário domínio = 0, ruído = 0 e ordem de reação química = 1, o que será considerado a partir de agora como nosso domínio base das simulações. Nos outros cenários já consegue-se observar uma distribuição homogênea entre si, contendo 20 simulações cada. Ou seja, o banco de dados está estruturado majoritariamente dentro de um domínio, com uma quantidade discrepante de amostras entre o domínio base e os demais, indicando dessa forma que a rede terá uma quantidade menor de dados para aprender o comportamento das classes presentes nos domínios de 1 a 6.

Após a visualização de como as simulações estão distribuídas dentro de cada cenário, foi necessário entender o comportamento geral das variáveis presentes no banco, onde o objetivo foi compreender a escala das variáveis, sua variabilidade e desvio padrão. No entanto, devido à grande quantidade de dados em cada variável, foi necessário realizar uma condensação dos dados, como exemplificado no Capítulo 4. Para isso, cada variável, que antes continha 200 amostras por simulação, foi transformada em uma média, evidenciando qual o valor médio daquela variável em cada uma das simulações. Dessa forma, passamos a obter um banco de dados de 2860 linhas e 11 colunas. Depois dessa manipulação, foi possível verificar a variabilidade das variáveis entre as simulações, em que foram abordadas métricas como média, mediana, mínimos e máximos, e desvio padrão, exemplificados na Tabela 7.

Tabela 7 – Tabela de Dados Estatísticos.

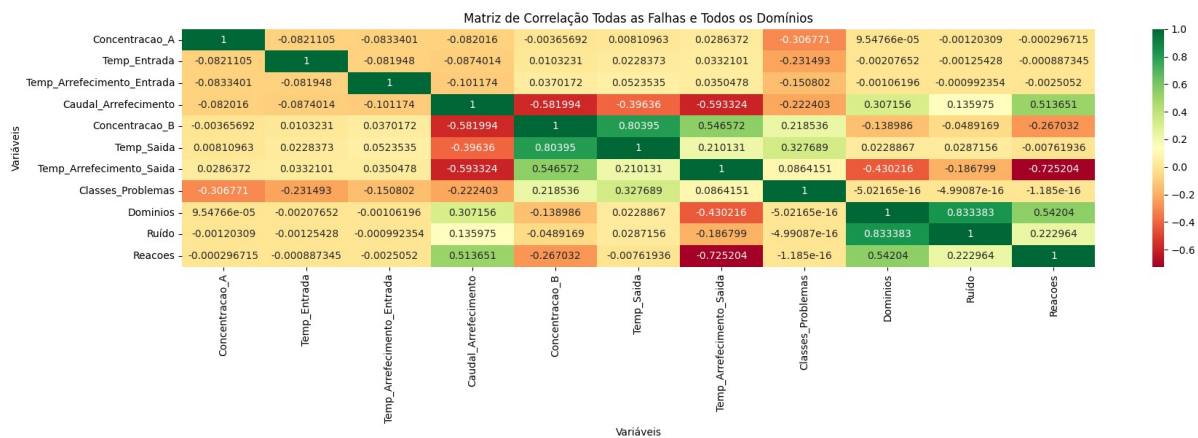
	Desvio Padrão	Mínimo	Máximo	Mediana
Concentração A (Ci)	0.13	0.89	1.49	1
Temperatura Entrada (Ti)	2.67	346.78	359.99	350.00
Temperatura Arrefecimento Entrada (Tci)	2.66	347.31	359.98	350.00
Caudal/Fluxo Arrefecimento (Qc)	0.18	0.0022	1.42	0.10
Concentração B (C)	9.52	355.65	448.76	430.91
Temperatura Saída (T)	8.10	355.50	435.76	416.76
Temperatura Arrefecimento Saída (Tc)	38.72	10.61	199.82	148.27

Como apresentado na Tabela 7, a escala das variáveis apresenta diferenças consideráveis entre si, inclusive em variáveis de mesma unidade de medida, como por exemplo Concentração de A e Concentração de B. Além disso, é observado que variáveis de arrefecimento do sistema possuem uma diferença de máximos e mínimos e um desvio padrão expressivos, indicando a possibilidade destas se mostrarem como as variáveis mais afetadas pelas falhas.

### 5.1.2 Correlação de Variáveis

A correlação linear é uma métrica utilizada para medir a correlação entre duas variáveis, ou seja, a relação entre elas. Com o objetivo de entender a correlação entre as variáveis do sistema, e o impacto dos cenários de teste, foi realizado o cálculo da matriz de correlação das variáveis presentes no banco. Na Figura 5 temos a matriz de correlação das variáveis demonstrada no formato de mapa de calor.

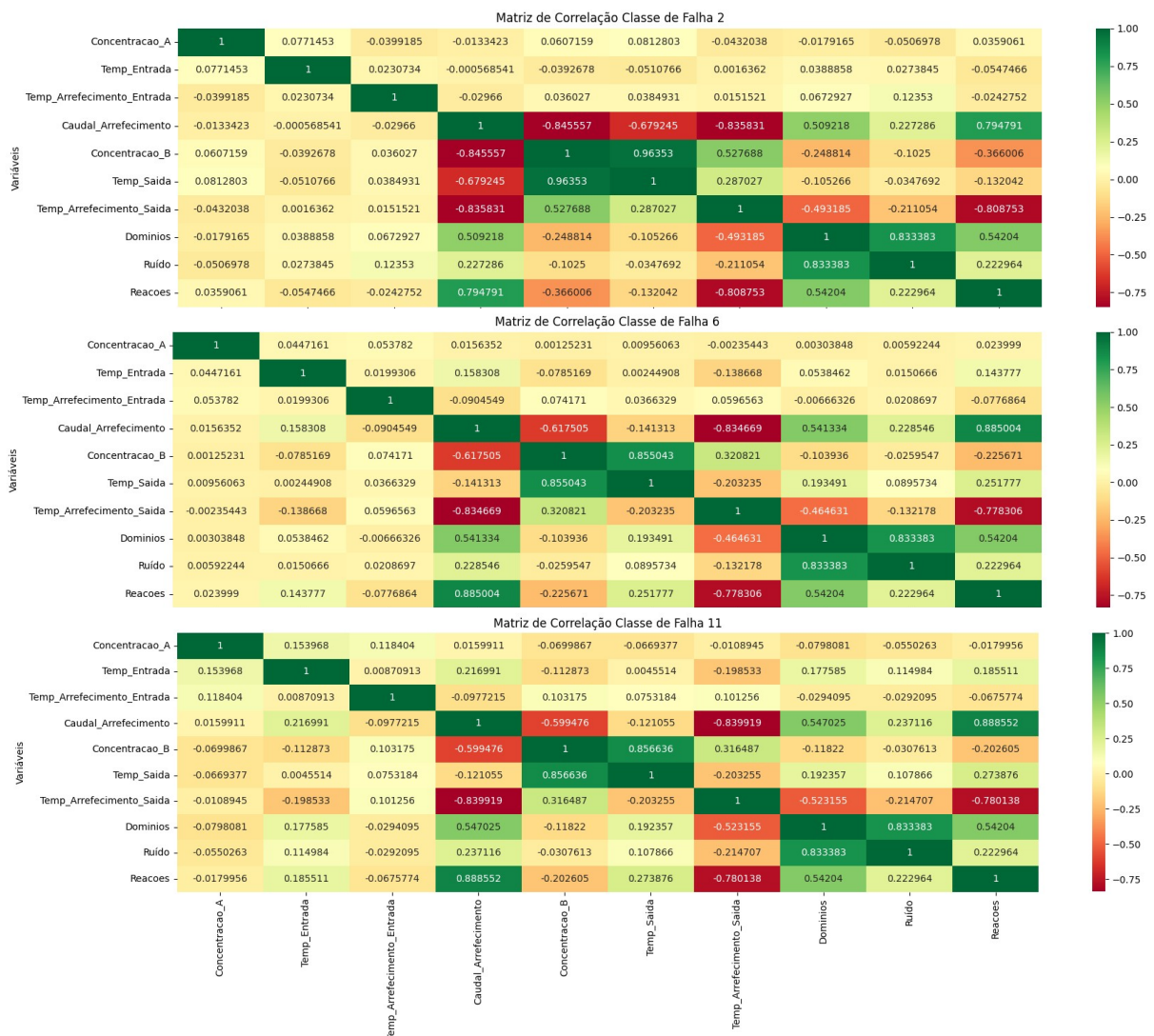
Figura 5 – Correlação das Variáveis do Sistema.



O mapa de calor da Figura 5 mostra que, no geral, a correlação entre as variáveis é baixa, existindo somente algumas relações proporcionais entre as variáveis de arrefecimento do sistema, e a concentração de B. Pode-se perceber, também, a baixa correlação das variáveis com classe de falha, domínio, ruído e ordem da reação, demonstrando que a mudança do cenário da simulação não tem uma relação direta com a mudança nos valores das variáveis do sistema. No entanto, devido a grande variabilidade de classes, domínios e ruídos, a baixa correlação apresentada na análise generalizada pode se apresentar como um falso resultado. Para garantir que as baixas correlações se apresentam também quando isolamos somente uma classe de falha, foram calculadas as matrizes de correlação para cada classe de falha, onde foram escolhidas 3 classes de falhas distintas para demonstração, uma falha de processo, uma de sensor e uma de distúrbio.

Na Figura 6 temos o mapa de calor da correlação das 3 classes de falha distintas, classe 2 - Falha de Processo, classe 6 - Falha se sensor e classe 11 - Falha de distúrbio.

Figura 6 – Correlação das Variáveis dentro da Classe de Falha 1, falha de processo.



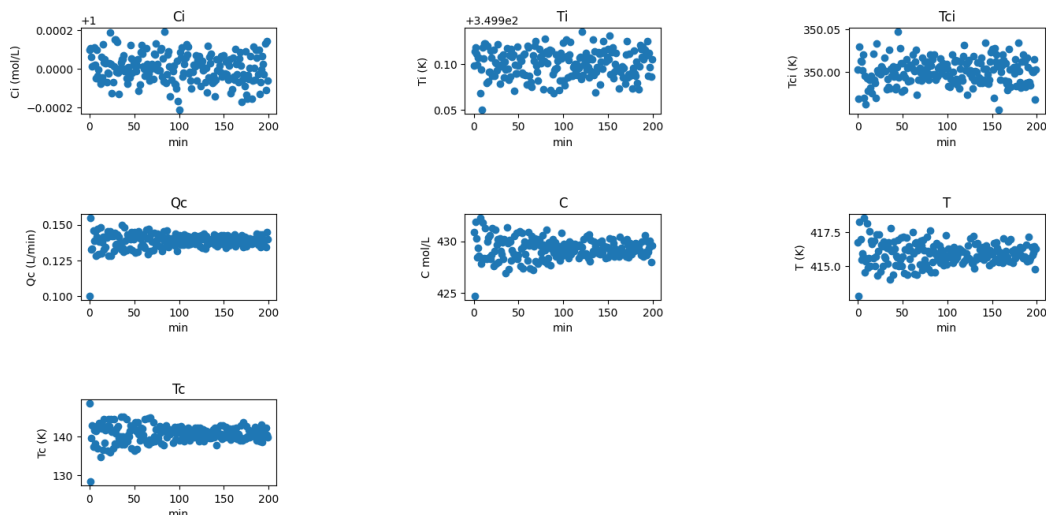
Pode-se observar que a falta de correlação entre as variáveis se manteve mesmo avaliando classes de falha específicas, dentro de três categorias diferentes de falha. A baixa correlação de variáveis indica uma dificuldade, ou impossibilidade, de se obter uma boa modelagem matemática do sistema, justificando dessa forma a utilização de algoritmos com capacidade de aprendizado, como a rede neural artificial.

### 5.1.3 Análise Comportamental das Variáveis

Nesta seção serão apresentadas as características das variáveis do sistema para cada uma das nove classes de falha consideradas neste trabalho, e para o funcionamento nominal do sistema. O objetivo é visualizar qual o comportamento observado em cada falha, a presença de valores discrepantes nas simulações, conhecidos como *outliers*. Um segundo objetivo é visualizar possíveis similaridades entre classes, pois demonstra a possibilidade de haver falsas classificações, onde a MLP acaba confundindo as classes de falha e classificando erroneamente as amostras.

Primeiramente, na Figura 7 temos sete gráficos onde estão presentes: Concentração de A, Temperatura de Entrada, Temperatura do Líquido de Arrefecimento na Entrada, Fluxo ou Caudal de Arrefecimento, Concentração de B, Temperatura de Saída e Temperatura do Líquido de Arrefecimento na Saída, cujas variáveis de representação foram apresentadas no Capítulo 3. Cada um dos gráficos demonstra o comportamento da variável ao longo dos 200 minutos de simulação.

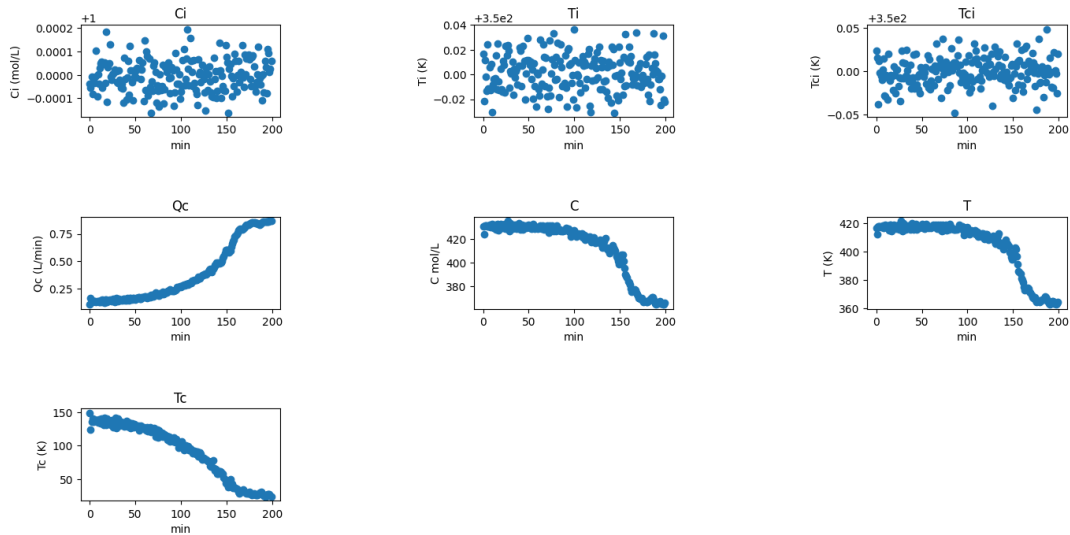
Figura 7 – Classe 0, Funcionamento Nominal do Sistema.



Na figura 7 observa-se que todas as variáveis do sistema possuem um comportamento estável, com pequenos distúrbio no início da simulação, demonstrando que não existem tendências ou variações abruptas do sistema em seu funcionamento nominal.

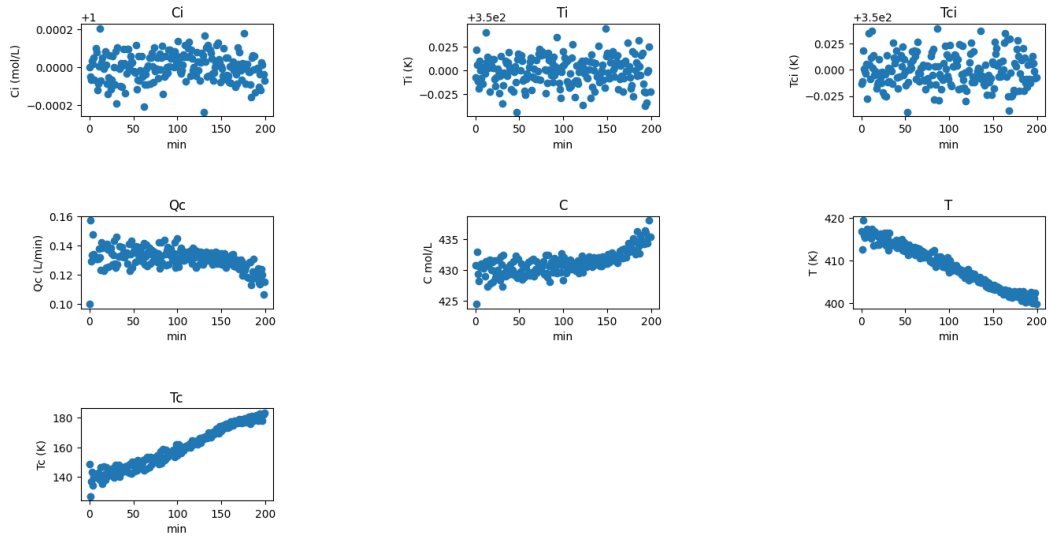
A seguir teremos imagens de cada classe de falha presente neste trabalho, apresentando os comportamentos das variáveis para cada tipo de falha, onde será feita uma comparação das variáveis em falha em relação ao comportamento nominal do sistema, elucidando as principais diferenças observadas entre a simulação em falha e a simulação nominal.

Figura 8 – Classe 1 - Falha na Catálise.



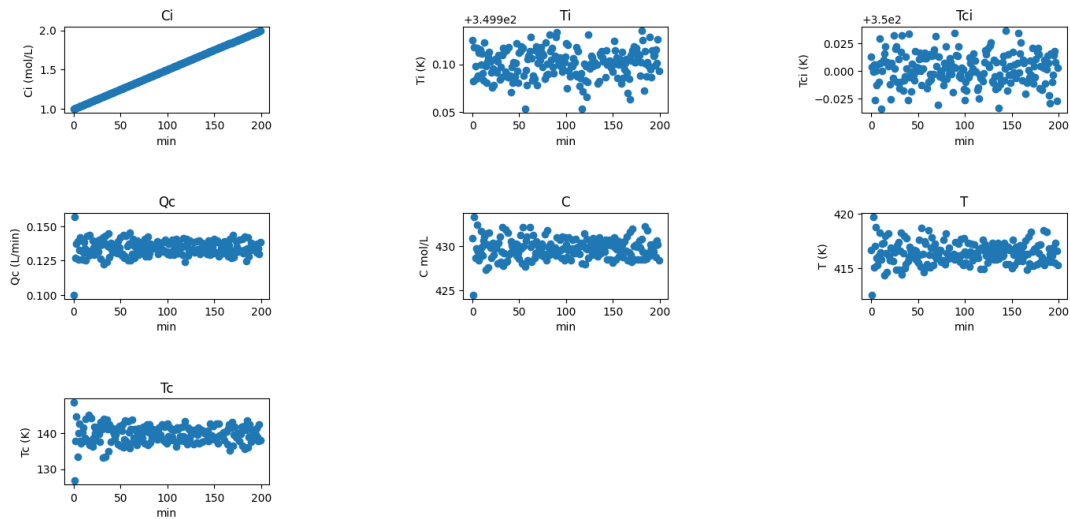
Para a falha de catálise, apresentada na Figura 8, ou classe de falha 1, pode-se observar que as variáveis  $C_i$ ,  $T_i$  e  $T_{ci}$  passaram a trabalhar em uma faixa de valores diferente da nominal, em que  $T_{ci}$  diminuiu seus valores nominais em mais de 1000 vezes. A variável  $Q_c$  apresentou uma mudança em sua tendência, apresentando agora uma tendência crescente ao longo da simulação, com uma faixa de valores superior que a nominal. As variáveis  $C$  e  $T$ , que antes apresentavam um comportamento estável, agora apresentam tendências de decréscimo. Além disso, a variável  $T_c$  além de agora apresentar um comportamento decrescente apresenta uma faixa de valores maior que a nominal. Portanto, vê-se que para a classe de falha 1, caracterizada como falha na catálise da reação, todas as variáveis apresentam mudanças quando comparadas com o funcionamento nominal do sistema e as variáveis  $Q_c$ ,  $C$ ,  $T$  e  $T_c$  apresentam mudanças nas faixas de valores consideradas normais.

Figura 9 – Classe 2 - Falha na Transferência de Calor.



Para a falha na transferência de calor, apresentada na Figura 9, pode-se observar que as variáveis  $C_i$ ,  $T_i$  e  $T_{ci}$  possuem valores e faixas de trabalho diferentes da nominal, sem apresentação de tendências ao longo da simulação. A variável  $Q_c$  apresentou uma mudança em sua tendência, apresentando agora uma tendência decrescente ao longo da simulação. As variáveis  $C$  e  $T$ , que apresentavam comportamento estável, agora apresentam tendências opostas, onde  $C$  aumenta ao longo do tempo e  $T$  decai. Além disso, a variável  $T_c$  além de agora apresentar um comportamento crescente, apresenta um comportamento linear.

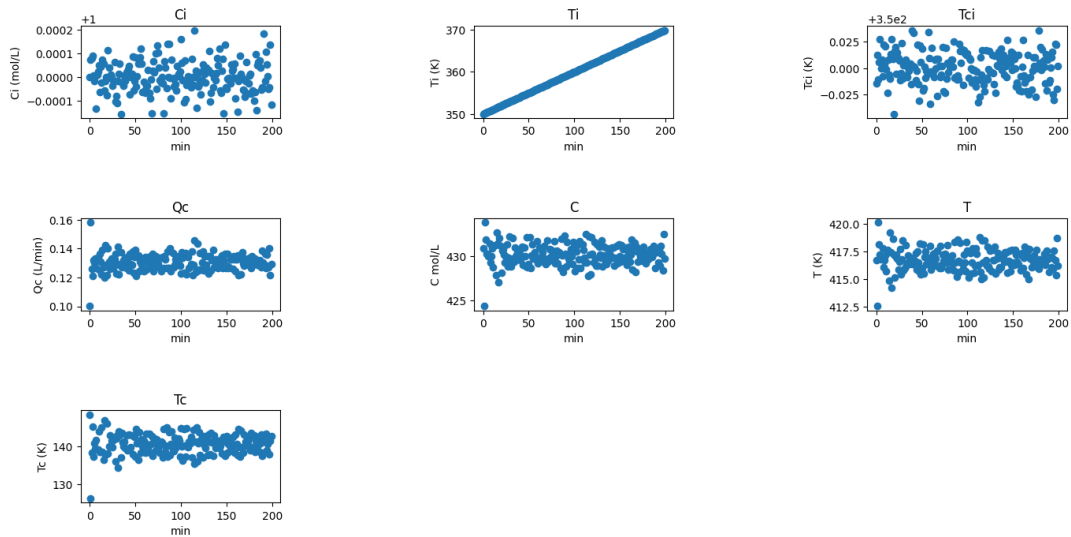
Figura 10 – Classe 3 - Distúrbio na Leitura do Sensor 1.



A falha de distúrbio na leitura do sensor 1, mostrado na Figura 10, altera completamente

o comportamento da variável  $C_i$ , em que passamos a ter um comportamento de uma rampa crescente, com valores em uma escala totalmente diferentes da escala presente no funcionamento nominal do sistema. É possível verificar que a variável  $T_{ci}$  passa a trabalhar na faixa  $-0.025 + 3.5e^{-5}$  (K) a  $0.025 + 3.5e^{-5}$  (K), reduzindo sua escala em mais de 1000 vezes. Enquanto as variáveis  $T_i$ ,  $Q_c$ ,  $C$ ,  $T$  e  $T_c$  apresentam comportamentos similares ao nominal.

Figura 11 – Classe 4 - Distúrbio na Leitura do Sensor 2.



Na Figura 11 é observado que a variável  $T_i$  apresenta um comportamento de rampa crescente e uma faixa de trabalho de 350(K) a 370(K), comportamento totalmente diferente do nominal. É possível verificar que, assim como na classe 3, a variável  $T_{ci}$  passa a trabalhar na faixa  $-0.025$  (K) a  $0.025$  K, reduzindo sua escala em mais de 1000 vezes. Enquanto as variáveis  $T_i$ ,  $Q_c$ ,  $C$ ,  $T$  e  $T_c$  apresentam comportamentos similares ao nominal.

Na falha de distúrbio na leitura do sensor 3, mostrado na Figura 12, observa-se que a variável  $T_{ci}$  adquire um comportamento de rampa crescente, enquanto a variável  $T_i$  apresenta uma faixa de trabalho diferente da nominal, variando de  $-0.025$ (K) a  $0.025$ (K). As outras variáveis apresentam comportamentos similares ao nominal.

Figura 12 – Classe 5 - Distúrbio na Leitura do Sensor 3.

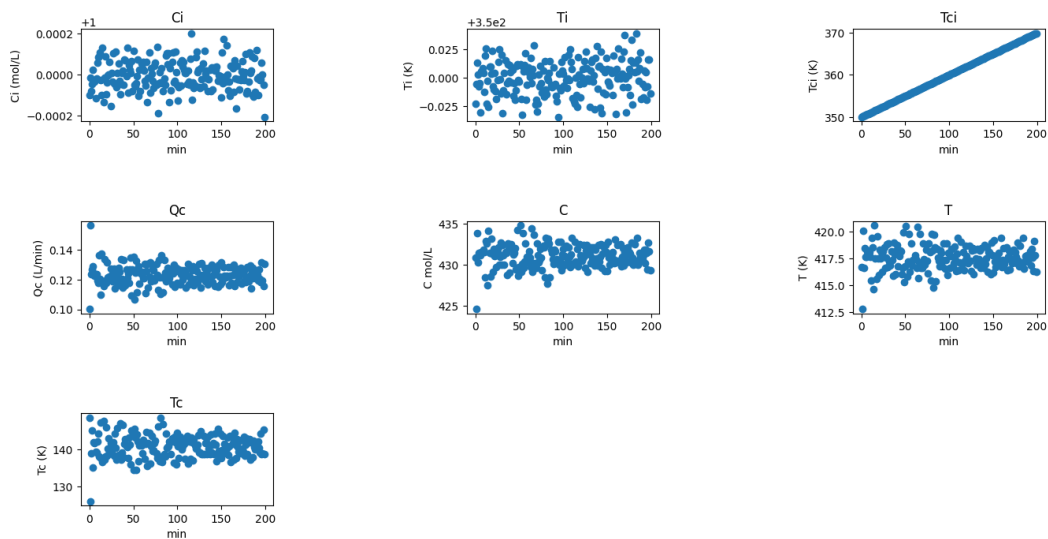
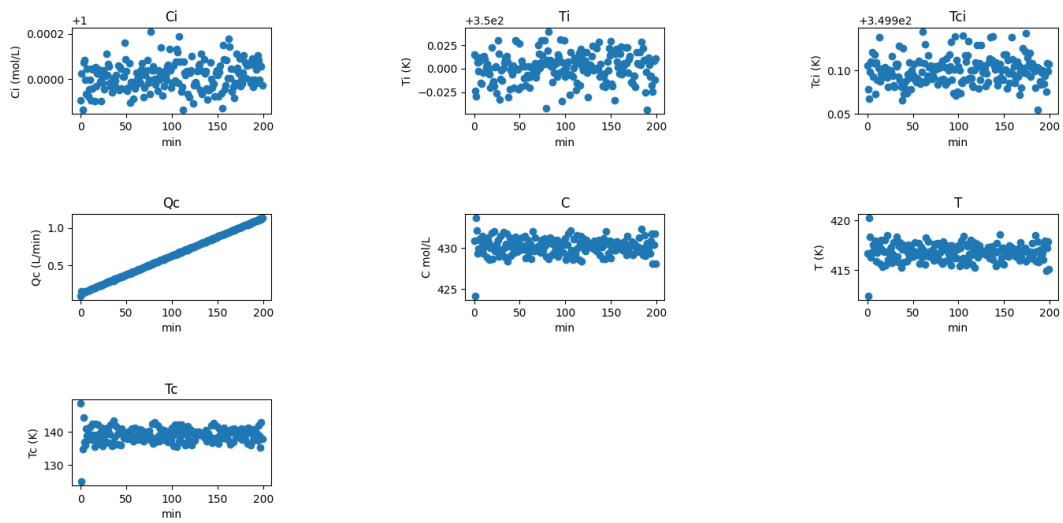
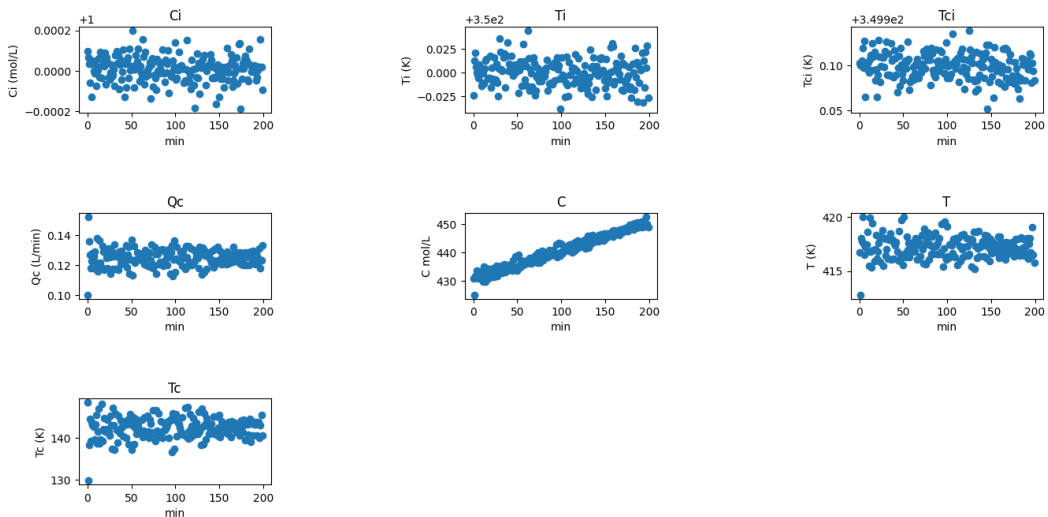


Figura 13 – Classe 6, Distúrbio na Leitura do Sensor 4.



Na falha de distúrbio na leitura do sensor 4 a variável Qc adquiriu comportamento de rampa crescente, a variável Ci reduziu sua faixa de trabalho para valores entre 0.0001(mol/L) a 0.0002(mol/L), a variável Ti aumentou sua faixa de trabalho para -0.025(K) a 0.025(K), a variável Tci apresentou redução de mais de 1000 vezes sua escala de trabalho. As demais variáveis do sistema apresentam comportamentos similares ao nominal

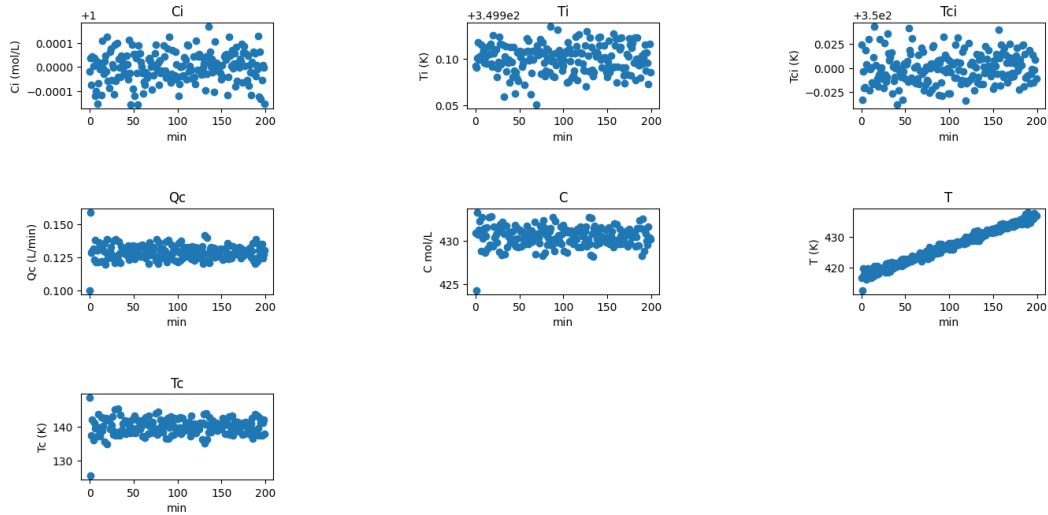
Figura 14 – Classe 7, Distúrbio na Leitura do Sensor 5.



Na Figura 14 vê-se que a variável C apresenta um comportamento de rampa crescente, sem grandes alterações na escala de trabalho de suas variáveis, as variáveis Ti e Tci apresentam as mes-

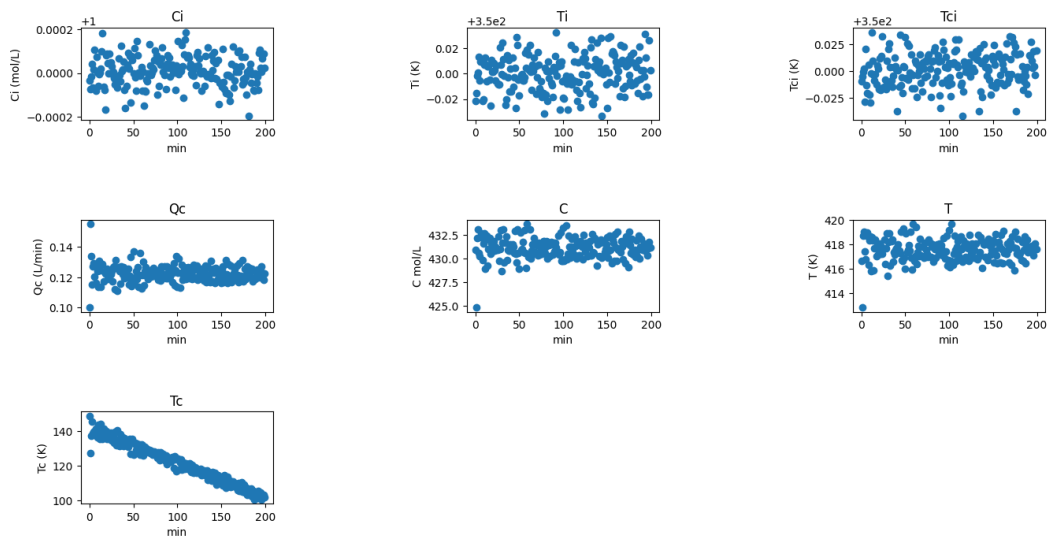
mas alterações que nas classes 5 e 6, enquanto as demais variáveis apresentam comportamentos similares ao nominal.

Figura 15 – Classe 8, Distúrbio na Leitura do Sensor 6.



Assim como para a classe 7, na Figura 15, para o comportamento de T, temos um comportamento de rampa crescente com presença de pequenos desvios ao longo da simulação, as variáveis Ci e Tci apresentam faixas de trabalho diferentes das nominais. As variáveis Ti, Qc, C e Tc apresentam comportamentos similares ao nominal.

Figura 16 – Classe 9, Distúrbio na Leitura do Sensor 7.



Na Figura 16 é possível observar que a classe 9 possui um comportamento de rampa decrescente com pequenos desvios ao longo da simulação, a variável Tci apresenta redução

em sua faixa de trabalho em mais de 1000 vezes a sua escala de trabalho. As demais variáveis apresentam comportamentos similares aos nominais.

A partir da comparação do comportamento nominal e das 12 classes de falha observa-se que as falhas de sensor todas apresentam impacto incisivo em uma ou mais variáveis, onde a variável atrelada ao sensor que apresenta distúrbio de leitura passa a apresentar um comportamento de rampa crescente ou decrescente, em sua maioria sem desvios ao longo da simulação. As falhas de processo são as falhas que apresentam impacto em grande parte das variáveis, apresentando aparecimento de tendências de crescimento, decrescimento e mudanças de faixa de trabalho da maior parte das variáveis estudadas.

## 5.2 Treinamento e Testes da MLP

A presente seção tem como objetivo apresentar e analisar os resultados obtidos na aplicação de uma rede neural Multilayer Perceptron na classificação de falhas em um reator de tanque agitado. Como apresentado no Capítulo 4 e visualizado no Capítulo 5, em 5.1.3, para este trabalho temos 9 classes de falhas, com diversos domínios de simulação. O resultado será subdividido em dois ambientes de testes. O primeiro considera o treinamento e teste da rede somente no domínio base da simulação, e o segundo considera treinamento e teste em todos os domínios disponíveis no banco, onde ao final será apresentada uma análise comparativa do desempenho nos dois ambientes.

Devido a estrutura das redes neurais e seus cálculos de pesos, já apresentados no Capítulo 3, a adição de neurônios e de camadas ocultas aumenta a acurácia de acertos da rede, visto que aumenta a capacidade de aprender relações mais complexas entre as variáveis, criando representações mais abstratas dos dados. No entanto, o aumento excessivo de neurônios e de camadas ocultas, pode acarretar em um *overfitting* e até mesmo na redução da acurácia. Uma vez que encontrada a melhor combinação de neurônios e de camadas ocultas, a adição de mais componentes na rede ocasiona novos distanciamentos nos cálculos.

Visto que a adição de neurônios e de camadas ocultas aumenta a acurácia de acertos da MLP, para garantir a melhor combinação possível de parâmetros, foram realizados testes com diferentes quantidades de neurônios nas camadas ocultas e de camadas ocultas nos dois ambientes de testes. No entanto, vale ressaltar que a quantidade de neurônios na camada de entrada e na camada de saída são proporcionais a quantidade de variáveis de entrada e de classes de falha. Portanto, temos um valor fixo de 7 neurônios na camada de entrada e 10 neurônios na camada de saída.

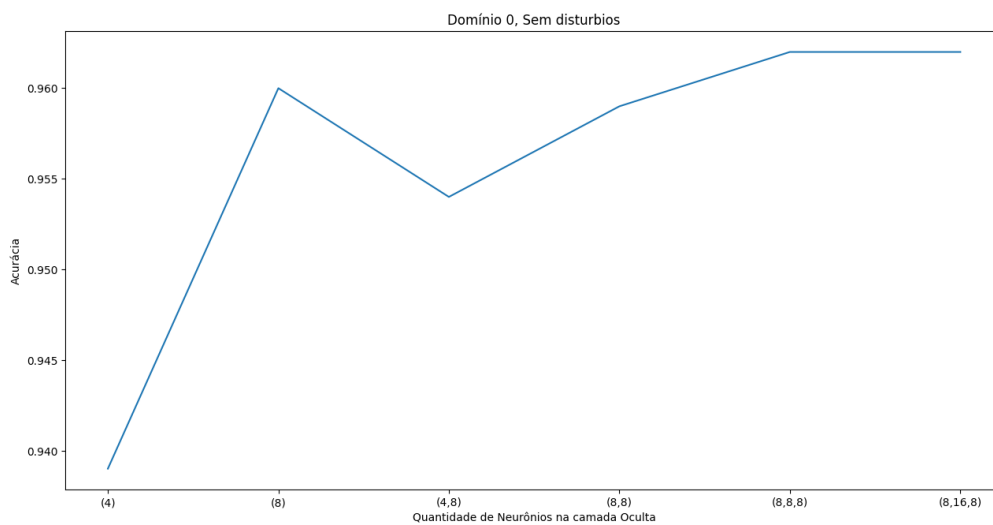
### 5.2.1 Treinamento e Teste no Domínio Base

Como realizado anteriormente na análise exploratória dos dados, o domínio base da simulação possui a maior concentração de amostras de falhas simuladas, totalizando 20000

amostras de funcionamento nominal e 20000 amostras de cada classe de falha. Portanto, para este problema de domínio único, espera-se uma maior facilidade na classificação das falhas quando relacionado com a classificação em multidomínios.

As acurácias dos testes realizados com cada configuração de neurônios está demonstrada na Figura 17, onde temos a informação de neurônios na camada oculta disposta no formato (neurônios camada 1, neurônios camada 2, ..., neurônios camada n).

Figura 17 – Acurácia em relação à quantidade de neurônios nas camadas ocultas.



Pode-se observar que existe um crescimento na acurácia da rede conforme a quantidade de neurônios nas camadas ocultas aumenta. Contudo, esse crescimento não ocorre de forma linear e existe um decréscimo ao acrescentar uma nova camada oculta na rede, demonstrando uma dificuldade de convergência quando treinada com 4 neurônios na primeira camada oculta e 8 neurônios na segunda camada oculta. Além disso, após atingir a acurácia máxima com 8 neurônios em cada camada oculta tem-se uma estabilização, indicando que a rede atingiu sua máxima capacidade de aprendizagem para os hiperparâmetros escolhidos, conforme explicado no Capítulo 4.

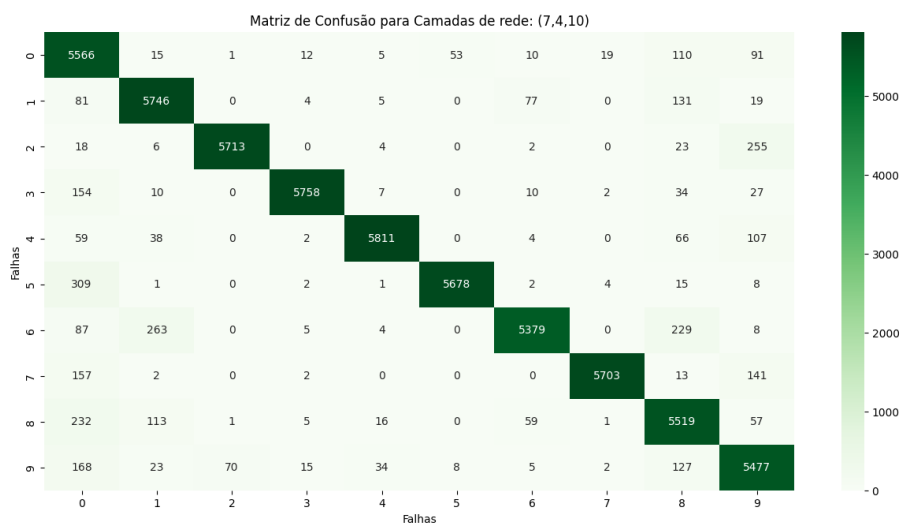
Apesar da acurácia ser uma boa métrica para entender a quantidade de erros e acertos de uma rede neural, ela não é capaz de demonstrar a quantidade de erros e acertos para cada classificação. Para conseguir visualizar de forma mais aprofundada o desempenho da rede serão demonstradas as matrizes de confusão de 4 principais configurações de neurônios nas camadas ocultas, que são: (4), (8), (4,8) e (8,8,8).

Como apresentado na seção 4, a matriz de confusão a tabela que cruza as classes verdadeiras com as classes previstas pelo modelo, oferecendo uma visão granular dos erros cometidos. Cada coluna da matriz representa uma classe prevista pela rede, e cada linha, uma classe do

sistema. Os valores dentro da matriz indicam a quantidade de exemplos que foram corretamente classificados (diagonal principal) e incorretamente classificados (fora da diagonal principal).

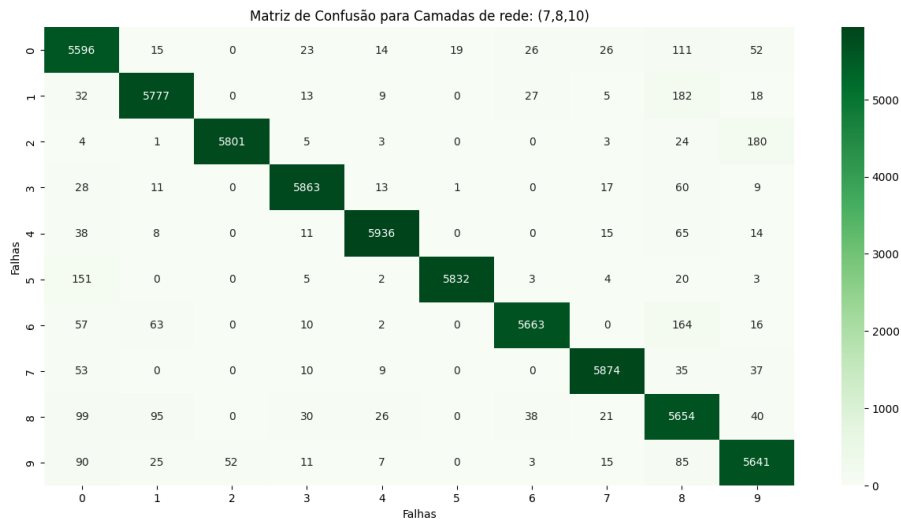
Para melhor visualização da matriz de confusão foi utilizado gráfico de mapa da calor como representação, em que quanto mais intensa a cor da célula, maior a quantidade de classificações na mesma, quanto mais branda a cor menos classificação foram executadas. Nota-se que quanto maior a presença de erros, mais cores intensas são apresentadas fora da diagonal principal.

Figura 18 – Matriz Confusão - 4 neurônios na camada oculta.



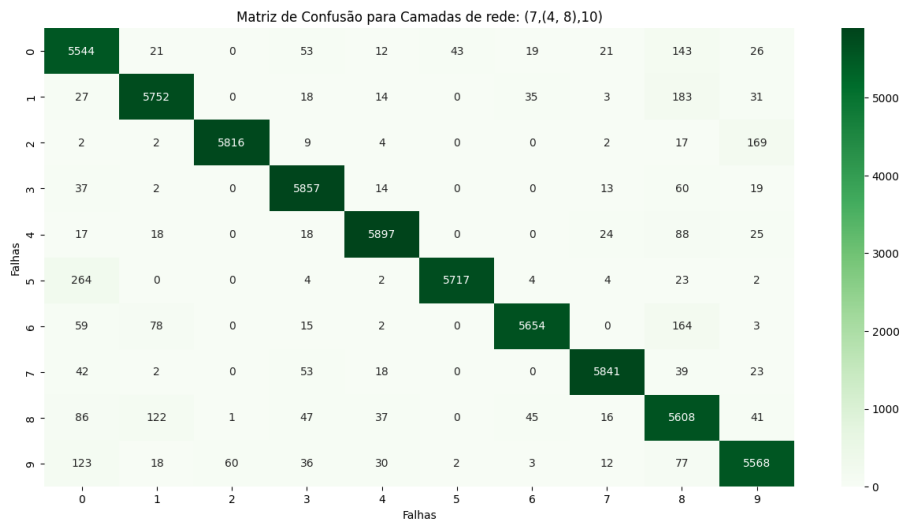
Na Figura 18 é possível visualizar que as classes 0, 1, 8 e 9 obtiveram a maior quantidade de classificações errôneas. A classe 0 apresentou confusão de classificação, em sua maioria, com as classes relacionadas as falhas de distúrbio na leitura de sensor, em que as principais confusões ocorreram entre 0 e 3, 0 e 5, 0 e 7 e 0 e 8. A partir dos gráficos de comportamento das variáveis em falha podemos observar que as classes 0, 3, 5, 7 e 8 tem uma grande quantidade de variáveis com comportamento similares, como Ci, Ti, Qc, C e Tc. A classe 1 apresentou confusão com as classes 6 e 8, e a partir dos gráficos comportamentais podemos observar que ambas as classes 1 e 6 apresentam um aumento da variável Qc ao longo da simulação. As classes 0 e 8, 8 e 6 e 9 e 2 apresentam diversas variáveis com comportamentos similares, facilitando a confusão da rede na classificação.

Figura 19 – Matriz Confusão - 8 neurônios na camada oculta.



Pode-se observar na Figura 19 que houve uma melhoria na quantidade de acertos das classes 0, 1, 8 e 9 como um todo. Contudo, ainda é possível observar que a confusão entre as classes 0 e 5, 8 e 1, 8 e 6 e 9 e 2 apresentada na Figura 18 se manteve. Além disso, apesar de permanecerem em valores muito baixos, erros na classificação das classes 2 a 7 aumentaram ligeiramente.

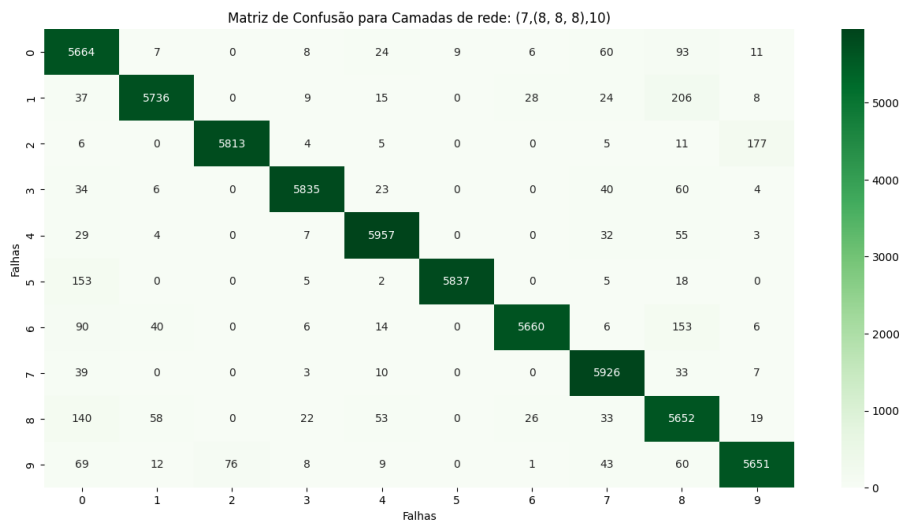
Figura 20 – Matriz Confusão - (4,8) neurônios na camada oculta.



Pode-se observar a partir da Figura 20 que houve uma piora na quantidade de acertos de forma geral entre as classes, mantendo a intensidade das confusões parecidas com o primeiro treinamento, demonstrado na Figura 18. Essa piora em relação ao treinamento anterior pode

ser explicado devido ao aumento na quantidade de camadas ocultas, em uma configuração que dificultou a convergência satisfatória da rede.

Figura 21 – Matriz Confusão - (8,8,8) neurônios na camada oculta.



Na figura 21 observa-se uma melhoria geral na acurácia de cada classe, no entanto ainda é possível observar as confusões, já vistas anteriormente, realizadas nas classificações da classe 0, 8 e 9, obtendo melhorias consideráveis nas confusões 1 e 8.

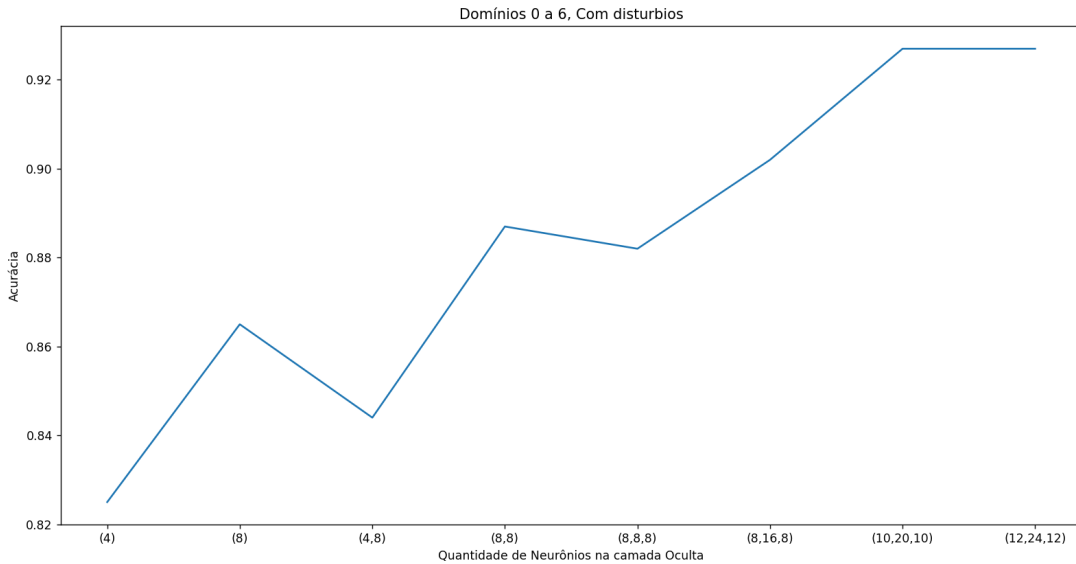
Ao realizar uma comparação dos 4 treinamentos verificamos uma melhoria na classificação das falhas em um contexto geral, reduzindo a quantidade de erros entre duas falhas específicas. No entanto, durante todo o treinamento vemos uma dificuldade da rede na diferenciação do funcionamento nominal do sistema (classe 0) com a classe 5, e também na diferenciação entre as classes 8 e 1, e 9 e 2, demonstrando uma dificuldade da rede em diferenciar estas classes mesmo com o aumento da quantidade de neurônios e camadas ocultas. Como um todo o desempenho da rede, em sua melhor configuração teve uma acurácia de 0,965, demonstrando 96,5 por cento de acertos quando somente um domínio é considerado.

### 5.2.2 Treinamento e Teste Todos os Domínios

Como explicado no 4 os domínios são mudanças nas configurações de variáveis do sistema, ocasionando uma ligeira mudança no comportamento das variáveis ao longo da simulação. Essas mudanças de domínio dificultam a classificação correta da rede, pois acrescentam variações comportamentais dentro de uma mesma classe de falha. Além disso, como já visto anteriormente na análise exploratória dos dados, a quantidade de amostras de falhas em cada domínio é distribuída de forma desigual, como apresentado na tabela 6. Por conta desse aumento na complexidade da classificação, e considerando o resultado obtido nos testes realizados considerando somente

um domínio, será realizado testes com maiores quantidades de neurônios e camadas ocultas em relação aos testes da seção anterior. A acurácia obtida nos testes está demonstrada na Figura 22.

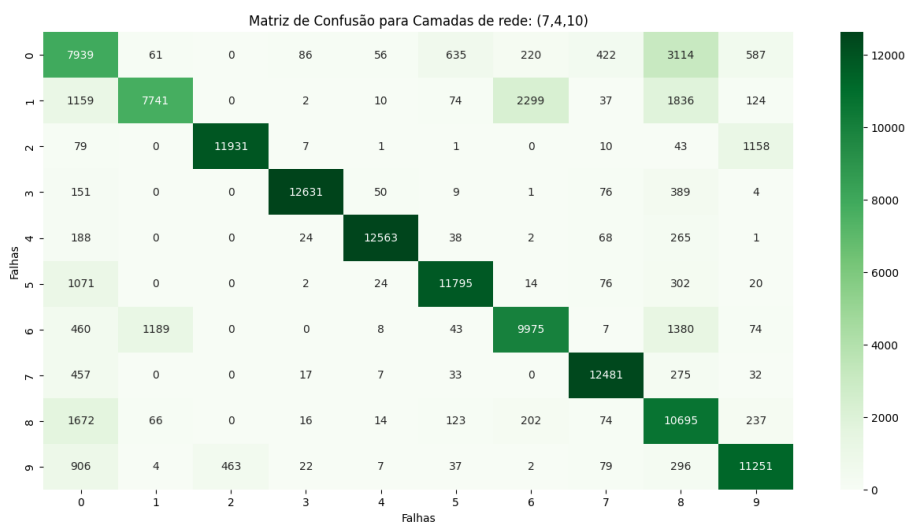
Figura 22 – Acurácia em relação à quantidade de neurônios e camadas ocultas.



Para a classificação das falhas no contexto multidomínio é possível confirmar a dificuldade na classificação de falhas quando comparamos as mesmas configurações utilizadas anteriormente, pois a acurácia alcançada se mostrou menor do que anteriormente.

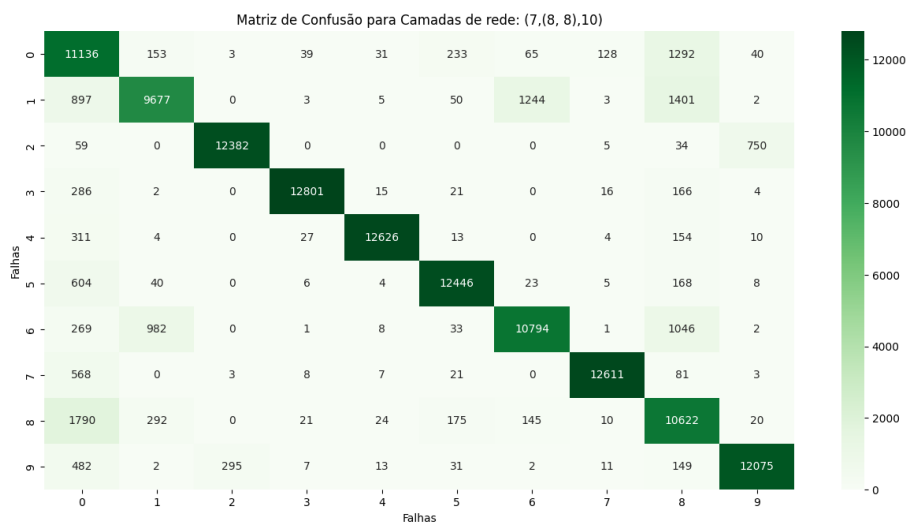
Assim como para a seção anterior, as matrizes de confusão de 4 configurações diferentes, (4), (8,8), (8,16,8) e (10,20,10), serão apresentadas e discutidas.

Figura 23 – Matriz Confusão - 4 neurônios na camada oculta.



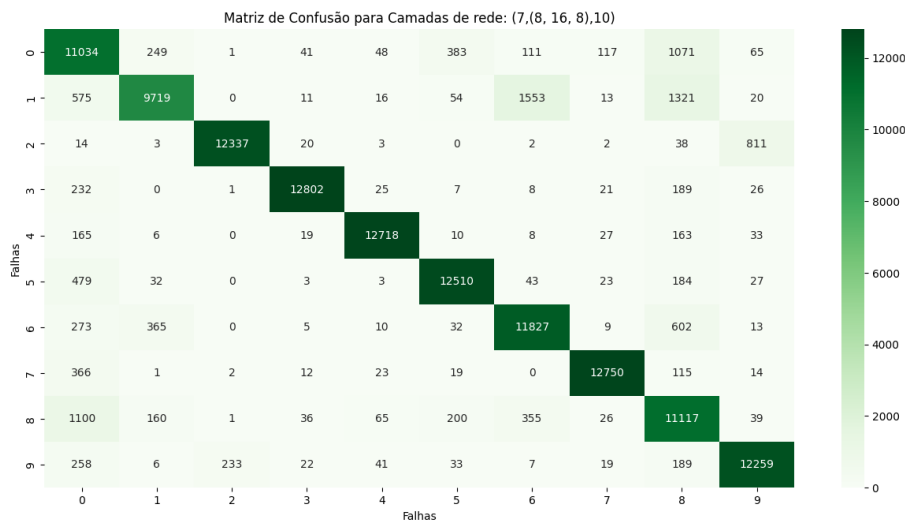
Ao analisar a Figura 23 é possível perceber uma quantidade expressiva de erros em diversas classes. A classe 0 apresentou confusões com 1, 5, 8 e 9, a classe 1 apresentou confusão com a classe 6, a classe 6 apresentou confusão com a classe 1, a classe 8 apresentou confusão com as classes 0, 1 e 6 e a classe 9 apresentou confusão com a classe 2. Diferentemente da seção anterior a explicação para as confusões não é mais baseada somente no comportamento similar das variáveis entre as falhas, mas também na possibilidade das classes apresentarem comportamentos similares em outros domínios. As confusões entre 0 e 8, 0 e 9, 1 e 6, 6 e 1, 8 e 0 não foram percebidas na Figura 18, portanto podem ser consideradas confusões que apareceram por conta de modificações realizadas nos domínios das simulações. Além disso, para as confusões 1 e 5, 8 e 1, e 9 e 2, já apresentadas na Figura 18 temos uma piora na quantidade de classificações errôneas, demonstrando que além das similaridades comportamentais já exemplificadas, o acréscimo de domínios diferentes também aumentou a dificuldade da MLP em classificar corretamente.

Figura 24 – Matriz Confusão do Treinamento com (8,8) neurônios na camada oculta.



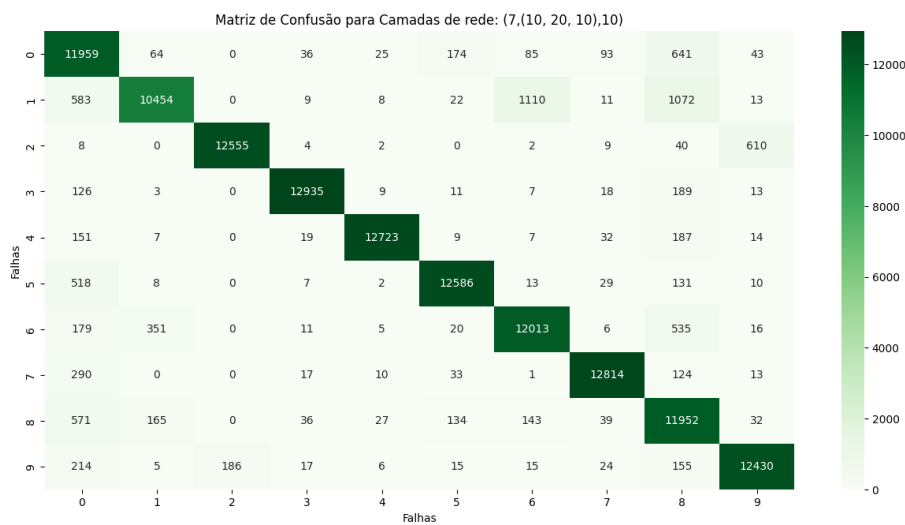
Na Figura 24 visualiza-se uma melhoria geral nas classificações, diminuindo a quantidade geral das confusões, mantendo uma confusão praticamente igual para 0 e 8. No entanto, apesar da melhoria geral, ainda é possível observar diversas confusões presentes também na Figura 18, apresentando melhorias significativas somente na confusões 2 e 9. Portanto, o aumento da quantidade de neurônios e camadas ocultas melhora para (8,8) a taxa de acerto da MLP como um todo, mas ainda não é capaz de diferenciar completamente as confusões apresentadas.

Figura 25 – Matriz Confusão do Treinamento com (8,16,8) neurônios na camada oculta.



A partir da Figura 25 é possível observar mais uma vez uma melhoria geral na quantidade de confusões executadas, mostrando melhor capacidade de diferenciar as classes 0 e 5, 6 e 1. Mas, ao comparar com a Figura 24, observamos que a dificuldade na diferenciação das classes 0 e 8, 6 e 1, 8 e 0, 8 e 1 e 9 e 2 ainda se mantém, sendo observado inclusive uma piora na diferenciação das classes 6 e 1. Portanto, a inclusão de uma camada oculta com 16 neurônios foi capaz de aumentar a acurácia como um todo, reduzindo a quantidade de erros cometidos, mas ainda apresentando dificuldade na diferenciação de classes específicas.

Figura 26 – Matriz Confusão do Treinamento com (10,20,10) neurônios na camada oculta.



Na Figura 26 vê-se uma melhoria considerável na diferenciação da classe 0 com as demais,

deixando de apresentar grandes confusões com outras classes. Essa melhoria também é observada na confusão entre 8 e 0. Apresentou também melhorias generalizadas na quantidade de erros apresentados em todas as classes, apesar disso ainda observamos a maior parte das confusões acontecendo entre 6 e 1, 8 e 0, 8 e 1, e 9 e 2.

Ao analisar as Figuras 23 a 26 conseguimos visualizar de forma clara a evolução do treinamento e as melhorias alcançadas através do aumento de neurônios e camadas ocultas. No entanto, assim como para a seção 1 onde existia somente um domínio, algumas falhas permanecem em todas as matrizes de confusão, demonstrando a dificuldade da rede em diferenciar falhas específicas. Como um todo o desempenho da rede MLP, em sua melhor configuração teve uma acurácia de 0,923, demonstrando 92,5% de acertos quando todos os domínios são considerados.

### 5.3 Análise comparativa

Ao realizar uma comparação entre os desempenhos da MLP na classificação das 9 falhas primeiramente em um único domínio e posteriormente em todos os domínios, pode-se visualizar o impacto da inserção dos domínios no problema, observando o aumento do esforço da rede em aprender os padrões de cada classe do sistema. Considerando a melhor configuração de ambos ambientes de testes tem-se que:

- Primeiro ambiente de teste: A rede conseguiu atingir uma acurácia de 96,5% ao ser configurada com 8 neurônios na primeira camada oculta, 16 neurônios na segunda camada oculta e 8 neurônios na terceira camada oculta. Apresentando dificuldade na diferenciação entre as classes 6 e 1, 8 e 1, 8 e 0 e 9 e 2.
- Segundo ambiente de teste: A rede conseguiu atingir 93,2% de acurácia quando configurada com 10 neurônios na primeira camada oculta, 20 neurônios na segunda camada oculta e 10 neurônios na terceira camada oculta. Apresentando dificuldade na diferenciação entre as classes 0 e 5, 8 e 1 e 9 e 2.

Portanto, o desempenho da MLP no primeiro ambiente de testes apresentou uma acurácia maior, com a utilização de um menor esforço computacional. Enquanto no ambiente 2 o desempenho da rede foi de 92,3% de acurácia com um maior esforço computacional. Observa-se que as confusões executadas entre 8 e 1, e 9 e 2 se mantém em ambos os cenários de teste, indicando que a dificuldade na diferenciação dessas classes ocorre independente da quantidade de domínios presente no sistema. No entanto é possível ver que a quantidade de confusões fixas no treinamento do ambiente 1 é maior que no ambiente de teste 2, indicando uma melhor capacidade de diferenciação geral do treinamento no segundo ambiente de teste.

Os resultados obtidos indicam que a rede neural proposta é capaz de classificar os diferentes tipos de falhas, especialmente quando treinada com dados de um único domínio de trabalho. Análises das matrizes de confusão permitiram identificar as classes mais difíceis de

se classificar e as principais fontes de erro. Ao comparar os resultados obtidos em diferentes ambientes de testes, observou-se que o desempenho da rede pode variar significativamente em função das condições operacionais.

## 6 CONCLUSÃO

A rede neural MLP demonstrou ser uma ferramenta eficaz na classificação de falhas em sistemas complexos e com múltiplos domínios de funcionamento, como o sistema CSTR analisado. Os resultados indicam que a MLP é capaz de aprender padrões complexos nos dados e aplicar para novas amostras, mesmo quando a complexidade do problema aumenta.

Ao comparar o desempenho da MLP em ambientes diferentes de teste, observamos que a inclusão de múltiplos domínios aumenta a dificuldade da tarefa de classificação e diferenciação entre classes. Isso acontece devido a necessidade da rede em aprender padrões com uma maior variabilidade nos dados, exigindo uma capacidade de generalização mais robusta.

Verifica-se que para este trabalho, as classes 8 e 1, assim como 9 e 2, apresentam uma maior dificuldade de classificação em ambientes diferentes, independente da quantidade de neurônios e camadas ocultas. Sugerindo assim que as classes possuem características semelhantes entre si, tornando a sua diferenciação mais desafiadora para o modelo.

Além disso, a quantidade desigual de amostras das classes de falhas entre os domínios demonstra dificultar o aprendizado da rede, dificultando o aprendizado de padrões em domínios com a menor quantidade de amostras.

Portanto, diante da capacidade demonstrada pela MLP em aprender padrões complexos e classificar diferentes tipos de falhas em sistemas reator de tanque de agitação contínua (CSTR, do inglês *Stirred Tank Reactor*), seu potencial como ferramenta para monitoramento contínuo em controladores tolerantes à falha torna-se evidente. Ao integrar a MLP a um sistema de controle tolerante a falhas, é possível detectar e classificar desvios no funcionamento nominal do sistema, identificando as falhas de forma automatizada e permitindo ao controlador adaptar seu laço de controle. Ao inserir a MLP como módulo FDI (do inglês, *Fault and Isolation Diagnosis*) do controlador, é possível manter o desempenho do sistema mesmo na presença de falhas.

### 6.0.1 Trabalhos Futuros

O campo de aplicação de inteligência artificial para classificação de falhas apresenta diversas possibilidades para futuras pesquisas. A partir dos resultados obtidos neste trabalho, algumas direções podem ser exploradas.

A primeira linha de investigação consiste em explorar a aplicação de metodologias de transferência de conhecimento e *fine-tuning*, acelerando o treinamento da rede e melhorando seu desempenho ao utilizar dados pré-treinados em grandes conjuntos de dados.

Neste trabalho, devido à baixa quantidade de amostras das classes de falha de distúrbio, não foi implementada a classificação de falhas relacionadas aos distúrbios no sistema. Uma linha possível consiste em explorar técnicas de aumento de dados para aumentar a quantidade de dados de distúrbio, possibilitando a aplicação de redes neurais para classificação dos mesmos.

Uma segunda linha de pesquisa é a possibilidade da aplicação de hiperparâmetros capazes de diferenciar as classes de distúrbio das demais mesmo em um cenário de poucas amostras para treinamento.

Um outro exemplo de investigação que tem potencial de trazer benefícios significativos para a indústria é a integração de MLPs em sistemas de controle, com a implementação de redes neurais como módulos de classificação e diagnóstico de falhas para controladores tolerantes a falhas.

Em resumo, as possibilidades de trabalhos futuros abrangem desde a exploração de técnicas de transferência de conhecimento e aumento de dados até a integração com controladores para predição e controle de sistemas em tempo real.

## REFERÊNCIAS

- AMIN, A. A.; HASAN, K. Mahmood-ul. Hybrid fault tolerant control for air–fuel ratio control of internal combustion gasoline engine using kalman filters with advanced redundancy. **Measurement and Control**, SAGE Publications Sage UK: London, England, v. 52, n. 5-6, p. 473–492, 2019. Citado na página 18.
- ANDRADE, A. C. C. Desenvolvimento de uma metodologia utilizando rede neural artificial na detecção e diagnóstico de falhas para válvula de controle pneumática. Universidade Federal do Rio Grande do Norte, 2021. Citado 2 vezes nas páginas 18 e 23.
- BENINI, F. A. V. Rede neural recorrente com perturbação simultânea aplicada no problema do caixeiro viajante. **São Carlos: Universidade de São Paulo**, 2008. Citado na página 25.
- CHHIPA, A. A. *et al.* Adaptive neuro-fuzzy inference system-based maximum power tracking controller for variable speed wecs. **Energies**, MDPI, v. 14, n. 19, p. 6275, 2021. Citado na página 18.
- DROZDOVITCH, V. Radiation exposure to the thyroid after the chernobyl accident. **Frontiers in endocrinology**, Frontiers Media SA, v. 11, p. 569041, 2021. Citado na página 16.
- FERREIRA, W. D. Detecção de imagens falsificadas baseada em descritores locais de textura e rede neural convolucional. 2020. Citado na página 25.
- GONÇALVES, W. M. **Detecção e diagnóstico de falhas em processos industriais baseado em modelagem data-driven sob a ótica da indústria 4.0**. Dissertação (B.S. thesis), 2022. Citado 4 vezes nas páginas 16, 18, 23 e 24.
- GONTIJO, M. C. A. *et al.* A produção científica sobre inteligência artificial e seus impactos: análise de indicadores bibliométricos e altmétricos. Universidade Federal de Minas Gerais, 2020. Citado na página 18.
- JÚNIOR, F. F. Aprendizado de máquina para a detecção e diagnóstico de anomalias em aplicações acionadas por motores elétricos. Universidade de São Paulo, 2022. Citado 4 vezes nas páginas 16, 18, 23 e 30.
- JUVÊNCIO, M. H. C. *et al.* Redes neurais e svm aplicadas no desenvolvimento de sensores virtuais e detecção e diagnóstico de falhas em processos reacionais complexos. Universidade Federal de Alagoas, 2023. Citado 5 vezes nas páginas 18, 23, 24, 25 e 30.
- KAGGLE. **CSTR Domain Adaptation and Fault Diagnosis**. 2024. Acessado em: 23 de agosto de 2024. Disponível em: <<https://www.kaggle.com/datasets/eddardd/continuous-stirred-tank-reactor-domain-adaptation>>. Citado na página 21.
- KANG, J.-L. Visualization analysis for fault diagnosis in chemical processes using recurrent neural networks. **Journal of the Taiwan Institute of Chemical Engineers**, Elsevier, v. 112, p. 137–151, 2020. Citado na página 18.
- LEARN, S. **MLPClassifier**. 2024. Acessado em: 10 de agosto de 2024. Disponível em: <[https://scikit-learn.org/stable/modules/generated/sklearn.neural\\_network.MLPClassifier.html#sklearn.neural\\_network.MLPClassifier](https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html#sklearn.neural_network.MLPClassifier)>. Citado 2 vezes nas páginas 29 e 30.

- MARTINS, O. V. S.; QUADROS, M. M.; MUROFUSHI, R. H.; LEITE, V. J. S. Controle híbrido adaptativo tolerante a falhas. In: **Anais do XXIV Congresso Brasileiro de Automática**, 2022. Citado na página 14.
- MONTESUMA, E. F. **Cross-Domain Fault Diagnosis through Optimal Transport**. Tese (Bachelor's Thesis) — Universidade Federal do Ceará, 2021. Citado na página 21.
- MONTESUMA, E. F.; MULAS, M.; CORONA, F.; MBOULA, F.-M. N. Cross-domain fault diagnosis through optimal transport for a cstr process. **ScienceDirect**, p. 1–6, 2022. Citado na página 14.
- MONTESUMA, E. F.; MULAS, M.; CORONA, F.; MBOULA, F. M. N. Cross-domain fault diagnosis through optimal transport for a cstr process. **IFAC-PapersOnLine**, Elsevier, v. 55, n. 7, p. 946–951, 2022. Citado 3 vezes nas páginas 21, 26 e 27.
- OGATA, K. **Engenharia de Controle Moderno**. São Paulo: Person Prentice Hall, 2010. ISBN 8576058103. Citado na página 15.
- PARK, Y.-S.; LEK, S. Artificial neural networks: Multilayer perceptron for ecological modeling. In: **Developments in environmental modelling**. [S.l.]: Elsevier, 2016. v. 28, p. 123–140. Citado na página 25.
- PILARIO, K. E. **MathWorks - Feedback-controlled CSTR Process for Fault Simulation**. 2023. <<https://www.mathworks.com/matlabcentral/fileexchange/66189-feedback-controlled-cstr-process-for-fault-simulation>>. Acessado em: 15 de Agosto de 2023. Citado 2 vezes nas páginas 21 e 27.
- PILARIO, K. E. S.; CAO, Y. Canonical variate dissimilarity analysis for process incipient fault detection. **IEEE Transactions on Industrial Informatics**, IEEE, v. 14, n. 12, p. 5308–5315, 2018. Citado 5 vezes nas páginas 18, 20, 21, 22 e 27.
- QUADROS, M. M.; LEITE, V. J.; PALHARES, R. M. Robust fault hiding approach for t–s fuzzy systems with unmeasured premise variables. **Elsevier**, 2022. Citado na página 14.
- QUADROS, M. M.; LEITE, V. J.; PALHARES, R. M.; BESSA, I. V. D. Fault tolerant control for linear parameter varying systems: An improved robust virtual actuator and sensor approach. **Elsevier**, p. 356–369, 2020. Citado na página 14.
- RUSSEL, S.; NORVIG, P. **Inteligência Artificial**. 3. ed. Rio de Janeiro: Elsevier Editora Ltda, 2013. Citado 2 vezes nas páginas 14 e 16.
- SCHNITMAN, L. **CONTROLADORES PREDITIVOS BASEADOS EM REDES NEURAIIS ARTIFICIAIS**. Dissertação (Mestrado) — Universidade Federal da Bahia, 1998. Citado na página 18.
- STEFFEN, T. **Control Reconfiguration of Dynamical Systems**. Berlin: Springer, 2005. ISBN 1610-7411. Citado na página 14.
- TELES, A. C.; FREITAS, A. C. de P.; RODRIGUES, C. A. Comparação de métodos de detecção de falhas em processos químicos utilizando inteligência artificial. **Periódico Tchê Química**, v. 16, n. 32, 2019. Citado na página 18.
- VIEIRA, M. V. Utilização de aprendizado supervisionado na previsão da demanda de energia no processo de produção de cumeno. Universidade Federal de São Carlos, 2021. Citado na página 18.