

**UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ**

**LUIS HENRIQUE FARIAS FERREIRA**

**SISTEMA DE DETECÇÃO DE FALTA EM MÁQUINA DE INDUÇÃO TRÍFASICA  
APLICANDO SISTEMAS INTELIGENTES**

**APUCARANA**

**2022**

**LUIS HENRIQUE FARIAS FERREIRA**

**SISTEMA DE DETECÇÃO DE FALTA EM MÁQUINA DE INDUÇÃO TRÍFASICA  
APLICANDO SISTEMAS INTELIGENTES**

**Fault detection system in three-phase induction machines applying intelligent  
systems**

Trabalho de conclusão de curso de graduação apresentada como requisito para obtenção do título de Bacharel em Engenharia elétrica da Universidade Tecnológica Federal do Paraná (UTFPR).  
Orientador(a): Thales Eugenio Portes de Almeida.  
Coorientador(a): Bruno de Nadai Nascimento.

**APUCARANA**

**2022**



Esta licença permite compartilhamento, remixe, adaptação e criação a partir do trabalho, mesmo para fins comerciais, desde que sejam atribuídos créditos ao(s) autor(es). Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.

**LUIS HENRIQUE FARIAS FERREIRA**

**SISTEMA DE DETECÇÃO DE FALTAS EM MÁQUINAS DE INDUÇÃO TRÍFASICA  
APLICANDO SISTEMAS INTELIGENTES**

Trabalho de Conclusão de Curso de Graduação  
apresentado como requisito para obtenção do título de  
Bacharel em Engenharia Elétrica da Universidade  
Tecnológica Federal do Paraná (UTFPR).

Data de aprovação: 24/Novembro/2022

---

Thales Eugenio Portes de Almeida  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Bruno de Nadai Nascimento  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Rodrigo Aparecido Fernandes Pereira  
Doutorado  
Universidade Tecnológica Federal do Paraná

---

Carlos Matheus Rodrigues de Oliveira  
Doutorado  
Universidade Tecnológica Federal do Paraná

**APUCARANA**

**2022**

## RESUMO

Este trabalho tem como objetivo desenvolver um sistema inteligente para o monitoramento de motores de indução, capaz de detectar curtos-circuitos de baixa intensidade no estator da máquina. Com isso diminuindo a ocorrência de manutenções corretivas em favor das preditivas, barateando assim os custos de operação da planta industrial, através da diminuição das paradas não programadas da linha de produção. Para treinamento do sistema inteligente foi criado um banco de dados através da simulação do comportamento de um motor em diferentes condições de operação, utilizando estes dados várias topologias de redes neurais foram treinadas, e tiveram seu desempenho analisado e comparado. Além disto o trabalho também discorre sobre o funcionamento de motores de indução, tipos de falhas e suas ocorrências em motores de indução industriais, o impacto dos curtos-circuitos no estator do motor, a indústria 4.0 e sua implementação no Brasil, as características e utilização de sistemas inteligentes e o funcionamento e treinamento de redes neurais.

Palavras-chave: motor de indução; detecção de curto-circuito; sistemas inteligentes; manutenção preditiva; indústria 4.0.

## **ABSTRACT**

The purpose of this paper is to develop an intelligent system for the monitoring of induction motors capable of detection low intensity short circuits in the machine's stator. Reducing the number of corrective maintenances in favor of predictive maintenances, therefore making the overall cost of the production plant cheaper, by reducing the occurrence of unscheduled stops in the production line. For that goal a database was created using the results of a computer simulation that emulates the behavior of induction machines under various conditions of fault, with that data various layouts of neural networks were trained, and their performance was analyzed and compared between each other. This work also discusses about how induction motors operate and the failures that can occur during their operation, the effects of short circuits in the motor's stator, the 4.0 industry and its implementation in Brazil, the characteristics and applications of intelligent systems and how neural networks work and how they can be trained.

Keywords: induction motor; short circuit detection; intelligent systems; predictive maintenance; 4.0 industry

## LISTA DE ILUSTRAÇÕES

Figura 1 - Motor de indução com rotor gaiola de esquilo em corte.....	14
Figura 2 - Gráfico da relação entre vida útil e temperatura de funcionamento.....	14
Figura 3 - Tipos de curto no estator do motor de indução .....	16
Figura 4 - Curto-circuito entre espiras.....	16
Figura 5 - Representação neurônio artificial .....	20
Figura 6 - Rede neural multicamada .....	22
Figura 7 - Resumo da utilização do sistema inteligente .....	23
Figura 8 - Representação perda .....	24
Figura 9 - Comportamento do motor, sem carga e em condições nominais .....	27
Figura 10 - Dados amostrados simulação sem carga e sem falta .....	28
Figura 11 - Dados amostrados simulação carga 5 Nm e sem falta.....	29
Figura 12 - Comportamento do motor, sem carga, com falta 4% .....	30
Figura 13 - Dados amostrados simulação sem carga, falta 4%.....	31
Figura 14 - Comportamento do motor, carga 3 Nm, falta 2%.....	32
Figura 15 - Dados amostrados simulação carga 3 Nm, falta 2% .....	33
Figura 16 - Gráfico sigmoide .....	36
Figura 17 - Precisão rede de seis camadas .....	37
Figura 18 - Perda rede de seis camadas.....	38
Figura 19 - Precisão rede de quatro camadas.....	39
Figura 20 - Perda rede de quatro camadas .....	40
Figura 21 - Precisão rede de três camadas .....	40
Figura 22 - Perda Rede de Três Camadas.....	41

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO .....</b>	<b>9</b>
<b>1.1</b>	<b>Justificativa .....</b>	<b>10</b>
<b>1.2</b>	<b>Objetivos.....</b>	<b>11</b>
1.2.1	Objetivo Geral .....	11
1.2.2	Objetivos específicos .....	12
<b>1.3</b>	<b>Estrutura do trabalho .....</b>	<b>12</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>13</b>
<b>2.1</b>	<b>Motor de indução.....</b>	<b>13</b>
<b>2.2</b>	<b>Falhas em motores de indução.....</b>	<b>15</b>
2.2.1	Curto-circuito no estator .....	15
<b>2.3</b>	<b>Inteligência artificial.....</b>	<b>18</b>
2.3.1	Redes neurais .....	20
2.3.1.1	Estrutura das redes neurais .....	21
2.3.1.2	Treinamento rede neural multicamada .....	22
2.3.1.3	Avaliação de desempenho das redes neurais .....	23
<b>3</b>	<b>SIMULAÇÕES DE FALTAS PARA CRIAÇÃO DO BANCO DE DADOS</b>	<b>26</b>
<b>3.1</b>	<b>Dados do Motor simulado .....</b>	<b>26</b>
<b>3.2</b>	<b>Resultados simulação .....</b>	<b>27</b>
3.2.1	Simulação motor sem carga e sem falta .....	27
3.2.2	Simulação do motor sem ocorrência de faltas e com carga.....	28
3.2.3	Simulação do Motor Sem carga e Com Ocorrência de faltas.....	29
3.2.4	Simulação do Motor Com carga e Com Ocorrência de faltas .....	31
<b>3.3</b>	<b>Banco de dados.....</b>	<b>33</b>
<b>4</b>	<b>DESENVOLVIMENTO DO SISTEMA INTELIGENTE.....</b>	<b>35</b>
<b>4.1</b>	<b>Estrutura da rede neural .....</b>	<b>36</b>
4.1.1	Rede de seis camadas.....	37
4.1.2	Rede de quatro camadas .....	38
4.1.3	Rede de três camadas.....	40
4.1.4	Comparação de classificação das estruturas .....	41
<b>4.2</b>	<b>Aplicação e constante desenvolvimento da rede neural .....</b>	<b>43</b>
<b>5</b>	<b>CONCLUSÃO.....</b>	<b>44</b>
	<b>REFERÊNCIAS.....</b>	<b>45</b>

## **APÊNDICE A - Código fonte para criação e treinamento da rede neural.52**



## 1 INTRODUÇÃO

Máquinas elétricas são essenciais para o funcionamento das indústrias a nível mundial, portanto se faz necessário que esses equipamentos apresentem grande confiabilidade, pois caso uma máquina venha a falhar, toda a linha de produção poderá ser interrompida, o que pode ocasionar grandes custos para a empresa. Desta forma a manutenção preditiva das máquinas é essencial, mas devido a recursos e mão de obra limitados, é interessante melhorar o planejamento e a previsibilidade da necessidade dessas manutenções. Com isso, o desenvolvimento e análise de técnicas para detecção de falhas e monitoramento de máquinas elétricas se torna cada vez mais importante.

Com a competição entre empresas e a constante busca por linhas de produção mais eficientes e menos custosas, cria-se a demanda pelo monitoramento contínuo da saúde dos motores. Fazendo uso de tal monitoramento é possível detectar de forma precoce falhas que possam vir a ocorrer no motor, permitindo assim medidas corretivas, com o intuito de aumentar a vida útil da máquina, e diminuir o tempo que a linha de produção precisaria ficar interrompida. Com isso diminuindo os custos da indústria e tornando-a mais competitiva perante o mercado (CHOUHDARY *et al.*, 2019).

Com o advento da indústria 4.0, técnicas como a análise em tempo real dos parâmetros de uma máquina estão tornando-se corriqueiros, isso devido a crescente capacidade de processamento e armazenamento de dados, além da maior conectividade dentro das indústrias. Com a utilização de big data os sistemas de inteligência artificial estão cada vez mais precisos, processando mais dados, criando mais oportunidades para industriais inteligentes (QI; TAO, 2018).

Considerando que cada vez mais os processos industriais estão sendo conectados através da internet das coisas, do inglês internet of things (IOT), o aproveitamento da estrutura já existente para possível análise do funcionamento das máquinas presentes na indústria, pode tornar o processo mais conectado e inteligente.

Em um país emergente como o Brasil, que possui um grande potencial de desenvolvimento industrial, um possível diferencial para a indústria interna seria o investimento e a rápida implantação de novas tecnologias associadas a indústria 4.0. Fazendo assim com que o Brasil se torne um dos países pioneiros na aplicação dessas tecnologias, tomando a frente da nova tendência que poderá revolucionar por completo as indústrias (SANTOS; MANHÃES; LIMA, 2018).

Os motores de indução são máquinas robustas que possuem uma longa vida útil, para máquinas de grande porte essa expectativa pode passar dos 20 anos, no entanto caso sejam operadas fora das condições previstas pelos fabricantes, tanto sua expectativa de vida quanto sua eficiência podem cair drasticamente, o que gera custos desnecessários para as indústrias (ANTONIO *et al.*, 2008). Monitorando os parâmetros de funcionamento do motor é possível manter sua eficiência, além de impedir que sua vida útil seja diminuída por fatores externos, possibilitando que manutenções preditivas sejam estruturadas, gerando o mínimo possível de impacto na linha de produção.

Considerando grandes linhas de produção, que possuem diversas máquinas, o monitoramento e análise individual das máquinas pode se mostrar inviável para a indústria convencional, mas com as técnicas de inteligência artificial e cloud computing, o esforço se torna consideravelmente menor, Pois a inteligência artificial facilita uma análise complexa e árdua, podendo tornar-se cada vez mais eficiente, considerando que seu banco de dados pode ser atualizado com os dados adquiridos dentro da própria indústria. Além disso, o cloud computing faz com que não seja necessário para a indústria possuir no local um sistema capaz de armazenar os dados adquiridos, facilitando e possivelmente diminuindo ainda mais os custos relacionados a implantação de sistemas para monitoramento dos parâmetros das máquinas e detecção de faltas.

O Brasil ainda possui grandes desafios para que as tecnologias da indústria 4.0 possam ser difundidas no país, facilitando assim a implementação das técnicas propostas no presente trabalho. De acordo com uma pesquisa feita pelo BNDES em 2016, 43% das empresas consultadas, não possuíam conhecimento sobre quais tecnologias digitais possuem maior potencial para impulsionar a competitividade nas indústrias, e de forma alarmante, foi constatado que apenas 1,6 % das empresas consultadas encontravam-se na geração 4.0 (VERMULM, 2018). Com isso, as técnicas de monitoramento de máquinas em tempo real também são pouco difundidas devida a sua necessidade de associação a tecnologias digitais presentes na indústria 4.0 para que seja viável sua implantação.

## **1.1 Justificativa**

Técnicas como o monitoramento de faltas em máquinas podem ser muito vantajosas para as empresas. A curto prazo com a diminuição do tempo em que a linha de produção precisa ser interrompida, aumentando assim a produtividade da indústria, e também a longo prazo, pois com a realização da manutenção preditiva nas máquinas, sua vida útil pode aumentar, fazendo

assim com que o índice de substituição de máquina se torne menor, gerando uma diminuição de custos de forma direta para a empresa.

Com a atual necessidade de as indústrias mundiais aderirem às tecnologias da indústria 4.0, investindo de forma estratégica para conseguir alcançar maior eficiência em sua produção, a inteligência artificial emerge como uma das maiores promessas da modernização das indústrias (LEE *et al.*, 2018).

A implementação da indústria 4.0 é um grande desafio para o Brasil. Além da dificuldade de engenharia para implantação das tecnologias da indústria 4.0, também é válido citar a barreira dos custos, pois como foi em todas as outras revoluções industriais sua aplicação vem atrelada a um custo financeiro. No entanto em uma pesquisa de 2016 chamada em inglês de *PWC 2016 Global Industry Survey*, prevê que industriais que estejam nos moldes da indústria 4.0 podem esperar uma redução de custos de aproximadamente 3,6 % anualmente, e um aumento na eficiência de 4,1 % também anualmente, além disso é esperado um aumento na qualidade da produção e um aumento na produtividade (GOLDMAN, F. L.; LAU, V. L.; MONTEIRO, 2020) (DAVIES; COOLE; SMITH, 2017).

Historicamente é observado que empresas que falharam em perceber e acompanhar os avanços tecnológicos acabaram perdendo espaço no mercado. Portanto, é interessante para a indústria brasileira que sejam desenvolvidas formas de aplicar essas novas tecnologias, fazendo com que as empresas possam implementar essas técnicas para obterem uma melhor eficiência e menor custo, podendo assim competir de forma mais vantajosa com empresas internacionais (DAVIES; COOLE; SMITH, 2017).

## **1.2 Objetivos**

### **1.2.1 Objetivo Geral**

Utilizar técnicas de sistemas inteligentes a fim de desenvolver um sistema que seja capaz de detectar faltas em máquinas de indução a partir da análise das formas de onda de tensão e corrente. Fazendo com que seja possível a realização de manutenção de caráter preditivo, aumentando a vida útil da máquina e diminuindo custos.

### 1.2.2 Objetivos específicos

Os objetivos propostos para esse trabalho são:

- Realizar um estudo crítico sobre redes inteligentes e sua aplicação em detecção de faltas
- Estudo dos fatores que podem causar faltas em motores
- Obter e analisar as formas de onda de tensão e corrente em máquinas de indução trifásicas por meio de simulações a fim de gerar um banco de dados
- Criar um sistema inteligente para detecção de faltas no estator de motores de indução trifásicos

### 1.3 Estrutura do trabalho

Este trabalho está dividido em cinco capítulos, onde o primeiro se refere a introdução da indústria 4.0 e a importância da detecção de faltas nas máquinas de indução.

No capítulo 2 será feita uma fundamentação teórica dos motores de indução e suas falhas e também sobre os sistemas inteligentes.

No capítulo 3 será apresentado as simulações de faltas para criação do banco de dados.

No capítulo 4 será discutido o desenvolvimento do sistema inteligente e seu banco de dados.

Por fim, no capítulo 5 serão apresentadas as conclusões do trabalho

## **2 FUNDAMENTAÇÃO TEÓRICA**

### **2.1 Motor de indução**

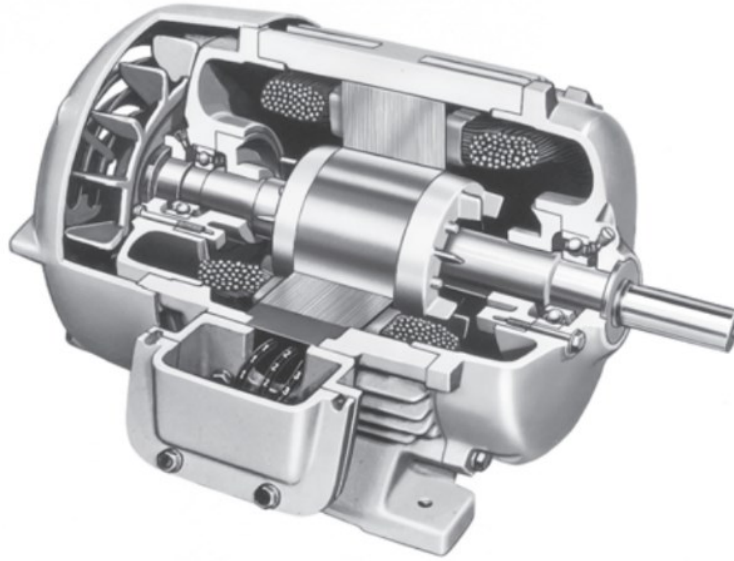
Máquinas de indução fazem parte das máquinas de corrente alternada assíncronas, se diferenciam das máquinas de corrente alternada síncronas pois sua corrente de campo é fornecida através de indução magnética, tendo seu princípio de funcionamento comparável ao de um transformador (CHAPMAN, 2013).

As máquinas de indução em âmbito industrial se resumem principalmente ao motor de indução, mais comumente o trifásico e em alguns casos específicos o monofásico. O motor monofásico é geralmente utilizado quando a indústria não possui alimentação trifásica, devido a sua menor eficiência e baixo conjugado de partida (LAVVA, 2021).

Em seu aspecto construtivo o motor de indução possui um estator idêntico àquele encontrado em máquinas síncronas, já seu rotor é de diferente construção podendo ser do tipo bobinado ou gaiola de esquilo, sendo que o rotor do motor mostrado na Figura 1 consiste em barras curto-circuitadas e é a configuração dominante no mercado (CHAPMAN, 2013).

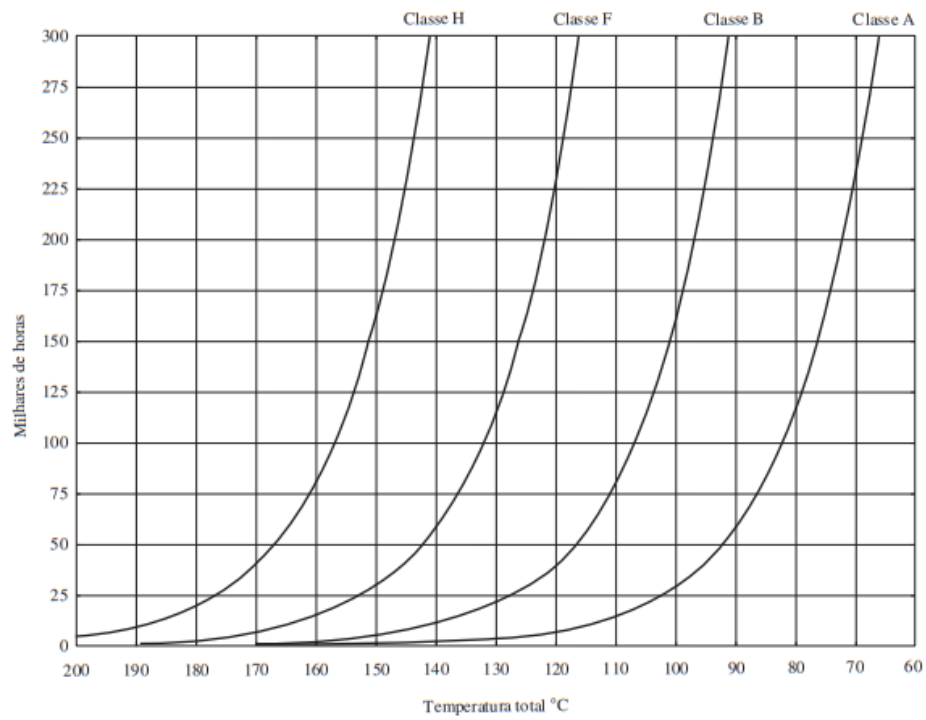
Os motores de indução possuem características térmicas, mecânicas e elétricas que devem ser respeitadas para seu funcionamento adequado e longevidade. Referente as características térmicas o motor de indução é dividido em categorias sendo elas A,E,B,F,H, sendo que a classe A permite uma elevação de temperatura de até 60 C°, e a categoria H permite uma elevação de até 125 C° (CHAPMAN, 2013). Como pode ser observado no gráfico apresentado na Figura 2, operar o motor acima da temperatura nominal da respectiva classe de isolamento gera uma perda significativa em vida útil da máquina.

**Figura 1 - Motor de indução com rotor gaiola de esquilo em corte**



**Fonte: Chapman (2013)**

**Figura 2 - Gráfico da relação entre vida útil e temperatura de funcionamento**



**Fonte: Chapman (2013)**

## 2.2 Falhas em motores de indução

As falhas em motores de indução podem ser causadas por diversos fatores internos e externos aos motores. As falhas no motor podem ser classificadas como falhas elétricas, ou falhas mecânicas, onde falhas elétricas por sua vez são divididas entre falhas no estator e falhas no rotor, e falhas mecânicas são as falhas de excentricidade e falhas no rolamento (SUETAKE, 2012).

Uma falha muito comum de se ocorrer são as falhas no estator, sendo que representam entre 30 a 40% das falhas em motores de indução. A principal causa das falhas no estator, é o curto-circuito entre enrolamentos de uma mesma bobina, essa falta ocorre geralmente devido a degradação da isolação dos enrolamentos, por si só não é um problema tão significativo, no entanto, caso essa falha passe despercebida, a mesma pode evoluir para um curto-circuito entre fases, ou um curto-circuito fase-terra, o que pode causar enormes danos na máquina (SHARIFI; EBRAHIMI, 2011).

A falha mecânica mais comum nos motores de indução é a falha nos rolamentos sendo responsável por aproximadamente 40% das falhas. Esta categoria de falha pode decorrer de vários fatores, podendo ser relacionada a vibrações causadas pela excentricidade, vibração dos mancais causada por drivers com chaveamento PWM, lubrificação entre outros fatores (SUETAKE, 2012). Sendo que a distribuição completa das falhas, pode ser vista na Tabela 1.

**Tabela 1 - Distribuição da ocorrência de falhas em motores de indução**

	Falha no rolamento	Falha no estator	Falha no rotor	Outros
EPRI	40%	38%	10%	12%
IEEE	42%	28%	8%	22%

Fonte: Vitor *et al.* (2014)

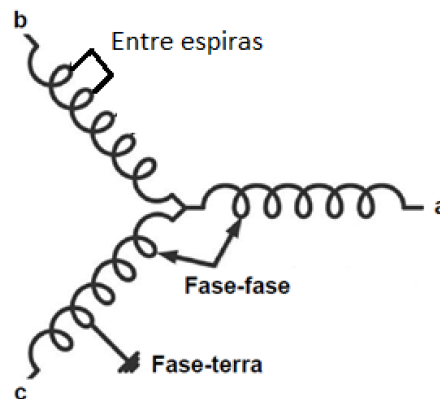
### 2.2.1 Curto-circuito no estator

A ocorrência de curto-circuito no estator, está entre os defeitos mais comuns de motores de indução (BAZAN, 2016). Este tipo de falta pode causar várias alterações no funcionamento do motor, como aumentos de temperatura, vibração e corrente, além da diminuição da eficiência da máquina, sendo que algumas destas alterações podem prejudicar mais ainda a isolação das espiras, podendo causar um curto-circuito mais severo (DUAN; ZIVANOVIC, 2012). Os curtos-circuitos no estator podem ser divididos em: curtos-circuitos entre bobinas de diferentes fases, curtos-circuitos entre bobinas de fase e a terra e pôr fim, o

objetivo de estudo deste trabalho, curtos-circuitos entre espiras de uma mesma bobina (BAZAN, 2016; EFTEKHARI *et al.*, 2014).

Na Figura 3 está sendo representado as possíveis ocorrências de curto no estator de um motor. Já na Figura 4 a situação exposta é especificamente um curto entre espiras de uma mesma bobina.

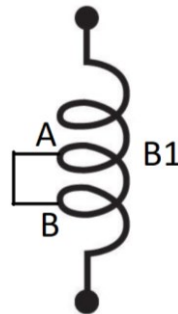
**Figura 3 - Tipos de curto no estator do motor de indução**



Fonte: Adaptado de Bazan (2016)

Para o desenvolvimento da simulação será analisada a influência do curto-circuito entre espiras de uma mesma bobina em 2 grandezas do estator, a resistência e a indutância.

**Figura 4 - Curto-circuito entre espiras**



Fonte: Autoria própria (2022)

A resistência elétrica pode ser dada conforme equação (1).

$$R = \rho * \frac{L}{A} \quad (1)$$

Sendo que R é a resistência elétrica ( $\Omega$ ),  $\rho$  é a resistividade elétrica do material ( $\frac{\Omega}{m}$ ), L é o comprimento do condutor e A é a seção reta do condutor.

Aplicando a equação (1) no curto-circuito representado na Figura 4, considerando que a resistividade e geometria do condutor são constantes e que a conexão entre as espiras em curto-circuito possui resistência igual a zero, pode-se afirmar então que A e B são eletricamente



o mesmo ponto, com isso o L entre eles é zero, ou seja entre A e B a resistência elétrica é nula. Portanto a resistência elétrica de uma bobina em curto-circuito, é sua resistência nominal menos a resistência decorrente das espiras que estão em curto-circuito.

$$R_{PC} = R_B - R_{ec} \quad (2)$$

Sendo que  $R_{PC}$  é a resistência do estator após o curto-circuito,  $R_B$  é sua resistência nominal, e  $R_{ec}$  é a resistência das espiras em curto-circuito.

A indutância total do estator  $L_T$  para aplicação no modelo dinâmico do motor, será considerada como a soma da autoindutância e da indutância mútua (VAS, 1998), sendo que a autoindutância é gerada pela corrente percorrendo a própria bobina, e a indutância mútua é criada pela interação entre a bobina e campos magnéticos adjacentes.

$$L_T = L_{11} + L_{1x} \quad (3)$$

A autoindutância pode ser calculada da seguinte maneira:

$$L_{11} = (N_1)^2 * \frac{\mu * \pi * A^2}{l_1} \quad (4)$$

Sendo que  $N_1$  é o número de espiras da bobina,  $\mu$  é a permeabilidade magnética ( $\frac{H}{m}$ ),  $A_1$  é a seção transversal ( $m^2$ ) e por fim  $l_1$  é o comprimento do núcleo (m).

Aplicando a equação (4) ao curto-circuito representado pela Figura 4, por simplicidade considera-se que a geometria da bobina não será alterada pela falta, com isso fica evidente que a variação na autoindutância da bobina será proporcional ao quadrado da variação do número de espiras.

$$\Delta L_{11} \propto \Delta(N_1)^2 \quad (5)$$

$$L_{11PC} = (N_1 - N_{cc})^2 * \frac{\mu * \pi * A^2}{l_1} \quad (6)$$

Sendo que  $L_{11PC}$  é a autoindutância da bobina com as espiras curto-circuitadas, e  $N_{cc}$  é o número de espiras em curto-circuito.

Já a indutância mútua da bobina, pode ser calculada conforme equação abaixo.

$$L_{1x} = N_1 * N_x * \frac{\mu * \pi * A^2}{l_1} \quad (7)$$

Sendo que  $L_{1x}$  é a indutância mútua na bobina 1, gerada pela interação do seu campo com aquele da bobina x, e  $N_x$  é o número de espiras da bobina x.

Fazendo as mesmas considerações utilizadas para calcular a autoindutância, chegamos novamente que o único parâmetro que irá se alterar, na indutância mútua da bobina curto-circuitada é o número de espiras da bobina 1.

$$\Delta L_{1x} \propto \Delta N_1 \quad (8)$$

$$L_{1xPC} = (N_1 - N_{cc}) * N_x * \frac{\mu * \pi * A^2}{l_1} \quad (9)$$

Sendo que  $L_{1xPC}$  é a indutância mútua na bobina 1, gerada pela interação do seu campo com aquele da bobina x após o curto-circuito.

Portanto após o curto-circuito a indutância total do estator será conforme abaixo:

$$L_{Tpc} = L_{11PC} + L_{1xPC} \quad (10)$$

Analisando as equações acima, fica claro que tanto a resistência quanto a indutância do estator irão diminuir após o curto-circuito.

### 2.3 Inteligência artificial

Inteligência artificial refere-se a um software, capaz de realizar tarefas que requisitariam da inteligência humana para serem completadas (SILVA; LENZ; FREITAS, 2018). Apesar de a inteligência artificial ser um campo estudado desde a década de 1950, nos últimos anos, tem se destacado, tanto no meio industrial quanto no dia a dia da população. Sua popularização pode estar associada a maior difusão de equipamentos digitais, a evolução das tecnologias diretamente relacionadas a inteligência artificial, e de forma complementar, ao

avanço de tecnologias paralelas que são utilizadas de forma adjunta com a inteligência artificial, como por exemplo *cloud computing*.

Os sistemas de inteligência artificial são capazes tanto de manipular dados, quanto manipular conhecimento, o que implica em utilizar novos conhecimentos ou conhecimentos já estabelecidos para resolução de problemas que não necessariamente precisam ser de caráter quantitativo (SILVA; LENZ; FREITAS, 2018).

Alguns dos modelos mais comuns para possibilitar o aprendizado automático de máquinas, do inglês *Machine Learning*, são: Lógica Fuzzy e Redes Neurais. A Lógica Fuzzy ou Lógica Difusa, é utilizada para sistemas que possuem informações imprecisas ou vagas, uma vantagem significativa da utilização da Lógica Fuzzy para *Machine Learning* é a possibilidade de sua implementação utilizando apenas regras linguísticas. As Redes Neurais têm como uma das principais vantagens a possibilidade da resolução de modelos complexos, sem uma definição precisa dos mesmos, e como o comportamento dinâmico de um motor em falta possui um modelo matemático não linear e complexo, foi escolhido a utilização do modelo de Redes Neurais.

Os métodos mais comuns para treinamento dos modelos de aprendizagem automática de máquinas são: Aprendizado Supervisionado, Aprendizado por Reforço e Aprendizado Não supervisionado, onde:

Aprendizado Supervisionado refere-se a técnicas de aprendizagem onde se possui um conjunto de entradas, e um conjunto de saídas referente a essas entradas. Sendo que o aprendizado ocorra no sentido de tentar minimizar o erro entre a saída encontrada pela máquina e a saída real do sistema (SILVA; LENZ; FREITAS, 2018).

Aprendizado por reforço consiste em fornecer um problema para máquina a qual a resolução depende de uma sequência de decisões, é definido um sistema de recompensa e penalidade para as escolhas da máquina, onde a máquina tenta maximizar sua recompensa diminuindo ao máximo as penalidades (FREITAS, 2019).

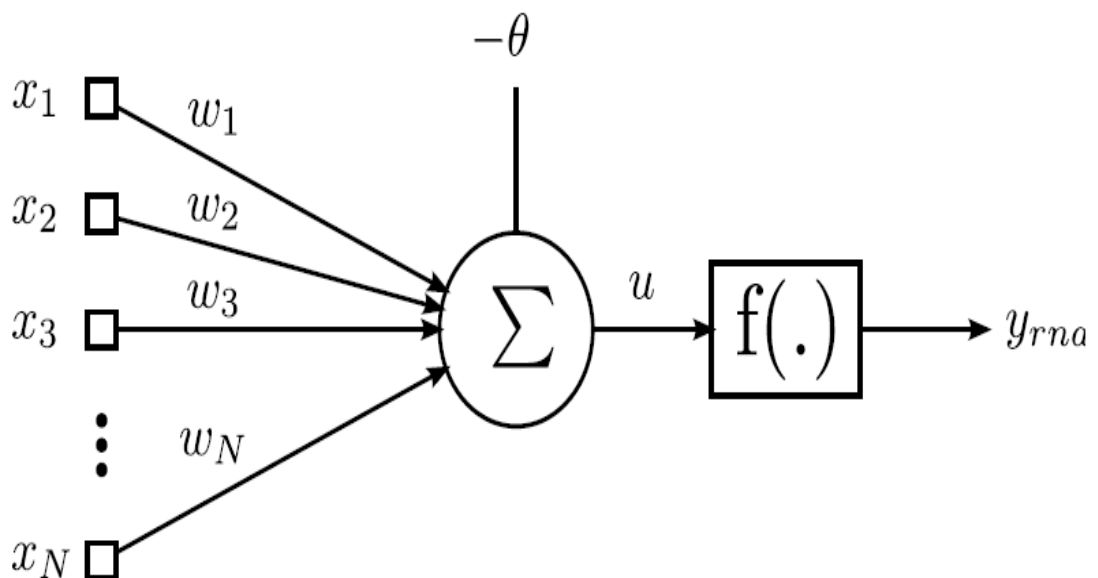
O Aprendizado não supervisionado consiste em um conjunto de dados de entrada de forma equivalente ao aprendizado supervisionado, no entanto nessa situação não se possui a saída referente a essas entradas, portanto a máquina vai procurar por conta própria padrões na entrada de dados, e sua saída será baseada na presença ou não de determinados padrões encontrados.

### 2.3.1 Redes neurais

Rede Neural Artificial (RNA), trata-se de um modelo computacional baseado nos neurônios e no funcionamento do cérebro humano. Redes neurais se baseiam em quatro princípios fundamentais, sendo eles: o processamento de linguagem natural para permitir que haja comunicação, representação do conhecimento para armazenamento, raciocínio automatizado para usar informações armazenadas e aprendizado de máquina para realizar adaptações diante de novas circunstâncias e criação de padrões (SILVA; LENZ; FREITAS, 2018).

As redes neurais funcionam de uma maneira em que existe uma ligação entre a entrada e saída, sendo que a ativação da entrada depende de fatores externos, onde esse fator para ativação é chamado de estímulo, sendo representado na Figura 5 o esquema de um neurônio artificial.

Figura 5 - Representação neurônio artificial



Fonte. Suetake (2012)

Como pode ser observado na Figura 5  $x_n$  é a entrada do neurônio, também chamado de estímulo de entrada do neurônio. O  $w_n$  se refere ao valor sináptico que cada estímulo terá, ou seja, o valor de  $w_N$  controla a influência de cada entrada no neurônio. O símbolo  $\Sigma$  é utilizado para representar uma combinação linear entre os estímulos de entrada e o valor sináptico. O parâmetro  $\theta$  é chamado de limiar de ativação e define o grau de ativação do resultado da combinação linear. O resultado da soma da combinação linear e do limiar de

ativação é dado por  $u$  que por sua vez é o argumento da função de ativação  $f(.)$  e seu retorno será  $y_{rna}$  (SILVA; SPATTI; FLAUZINO, 2010).

De forma resumida o funcionamento do RNA é da seguinte maneira, os estímulos de entrada  $x_n$  serão colhidos e serão ponderados por  $w_n$ , somados ao limiar de ativação  $\theta$ , que resultara em  $u$  que será a entrada da função de ativação  $f(.)$ , e sua saída será o resultado do neurônio artificial (SILVA; SPATTI; FLAUZINO, 2010), podendo ser definido matematicamente da seguinte maneira:

$$u = \theta + \sum_{i=1}^N w \cdot x_i \quad (11)$$

$$y_{rna} = f(u) \quad (12)$$

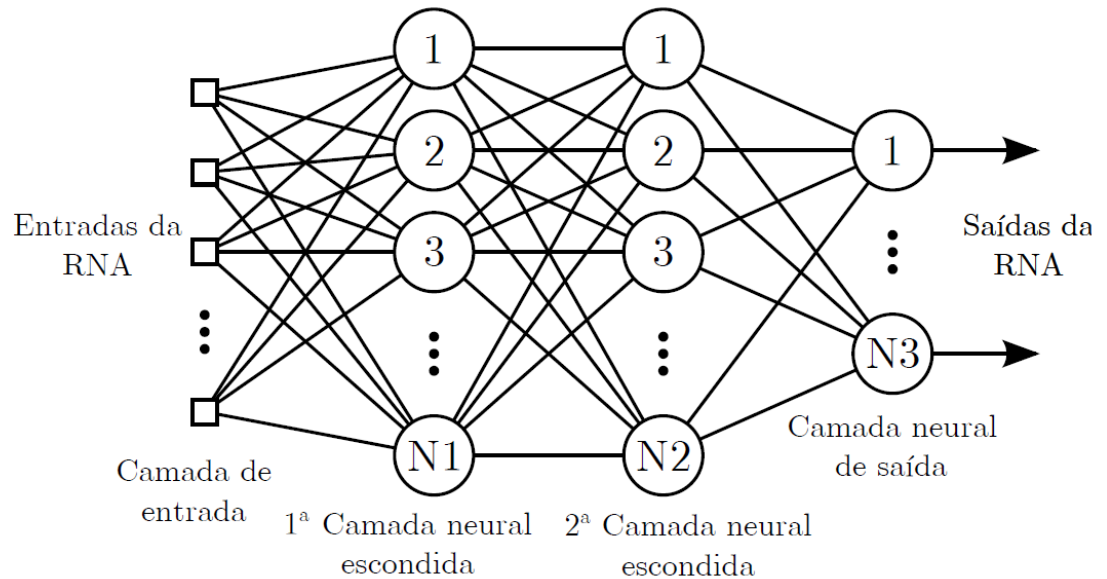
### 2.3.1.1 Estrutura das redes neurais

As duas estruturas mais comuns aplicadas nas redes neurais, são, a rede de neurônio simples, e a rede multicamadas. A rede neural simples é a forma mais básica possível para aplicação de uma rede neural, pois nessa configuração existe apenas um neurônio que fornece apenas uma resposta (SILVA; SPATTI; FLAUZINO, 2010).

A rede neural de multicamadas, é uma configuração em que grupos de neurônios artificiais possuem como suas entradas as saídas de um grupo de neurônios artificiais anterior. Fazendo com que se obtenha uma resposta mais precisa, mas também requer maior tempo de processamento da entrada (SILVA; SPATTI; FLAUZINO, 2010).

A representação de uma rede neural multicamada é mostrada na Figura 6, na qual podem ser identificados três diferentes tipos de camadas. A camada de entrada que será a responsável por receber a informação colhida e enviá-la para a próxima camada, mas ressaltando que a camada de entrada não realiza operações com os valores de entrada, sua saída será enviada para o segundo tipo de camada, chamada de camada oculta. Sendo que a camada oculta é responsável pelo efetivo processamento dos valores de entrada, até a chegada na última camada. A camada de saída é responsável por fornecer o resultado do processamento da rede (SILVA; SPATTI; FLAUZINO, 2010).

Figura 6 - Rede neural multicamada



Fonte. Suetake (2012)

### 2.3.1.2 Treinamento rede neural multicamada

A forma mais comum de se treinar uma rede neural multicamada é utilizando a regra Delta Generalizada. Esse é um processo de aprendizagem supervisionado, ou seja, para que essa técnica de treinamento seja aplicada, é necessário que se possua um conjunto de entradas, e que se saiba qual resposta esse determinado grupo de entradas deve produzir. O funcionamento da regra Delta Generalizada é da seguinte maneira:

*Fase Forward Propagation:* nessa fase um conjunto de entradas de treinamento são aplicadas na rede, esses dados são propagados e processados sem alteração dos valores sinápticos e limiares até se chegar à saída da rede, após isso os valores da saída da rede são comparados com os valores esperados, e a diferença entre esses valores chamado de desvio ou erro serão utilizados para ajustar os pesos e limiares da rede de neurônios (SILVA; SPATTI; FLAUZINO, 2010).

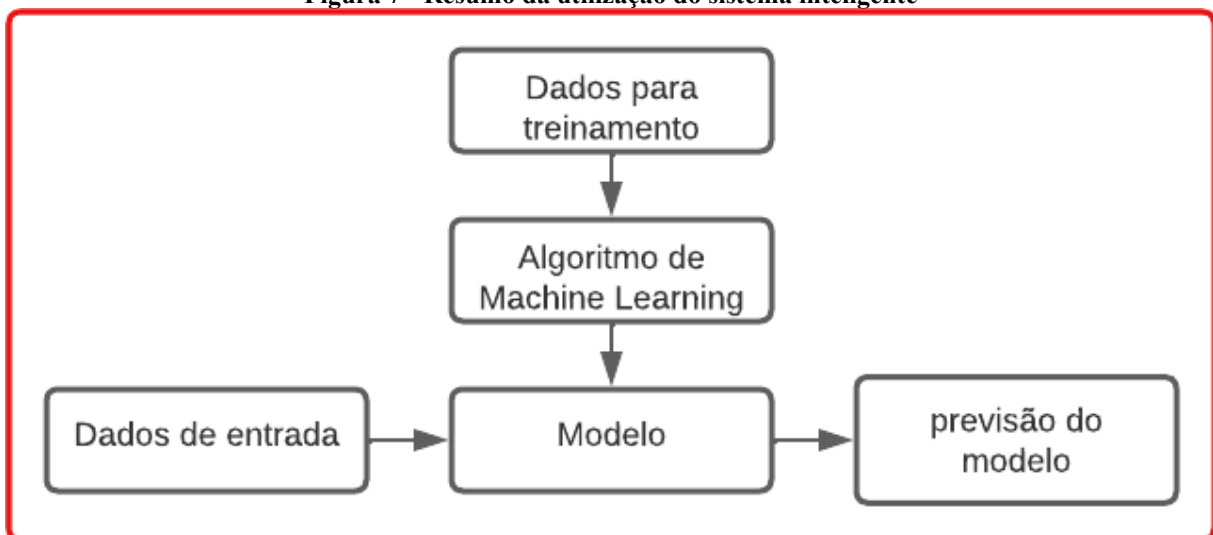
*Fase Backpropagation:* nessa fase, os valores de erros adquiridos na fase anterior serão utilizados para que os pesos e limiares dos neurônios sejam alterados, preparando o sistema para uma nova aplicação da fase *forward*, podendo esse processo ser repetido enquanto houver um conjunto de entradas e respostas conhecidas, ou até que se chegue a um valor de erro aceitável (SILVA; SPATTI; FLAUZINO, 2010).

Com o banco de dados formado, o processo de treinamento assistido pode ser iniciado, onde através de um serie de iterações utilizando as técnicas de *forward propagation* e

*backpropagation*, os vetores de valor sináptico e limiares serão ajustados, para que a rede possa responder da melhor maneira possível a entrada.

De forma resumida o treinamento e utilização da rede neural se dá como representado na Figura 7, onde “dados para treinamento” refere-se ao banco de dados que será utilizado no processo de treinamento assistido, “algoritmo de machine learning” é neste caso o algoritmo de regressão logística, “modelo” é a rede neural com seus pesos sinápticos e limiar de ativação ajustados pelo processo de treinamento que então será utilizada para processar os dados e chegar a uma conclusão, “dados de entrada” é entrada que deve ser processada para se chegar a uma conclusão, “previsão do sistema” é a conclusão que o sistema chega.

**Figura 7 - Resumo da utilização do sistema inteligente**



Fonte: Adaptado de Mesquita (2022)

### 2.3.1.3 Avaliação de desempenho das redes neurais

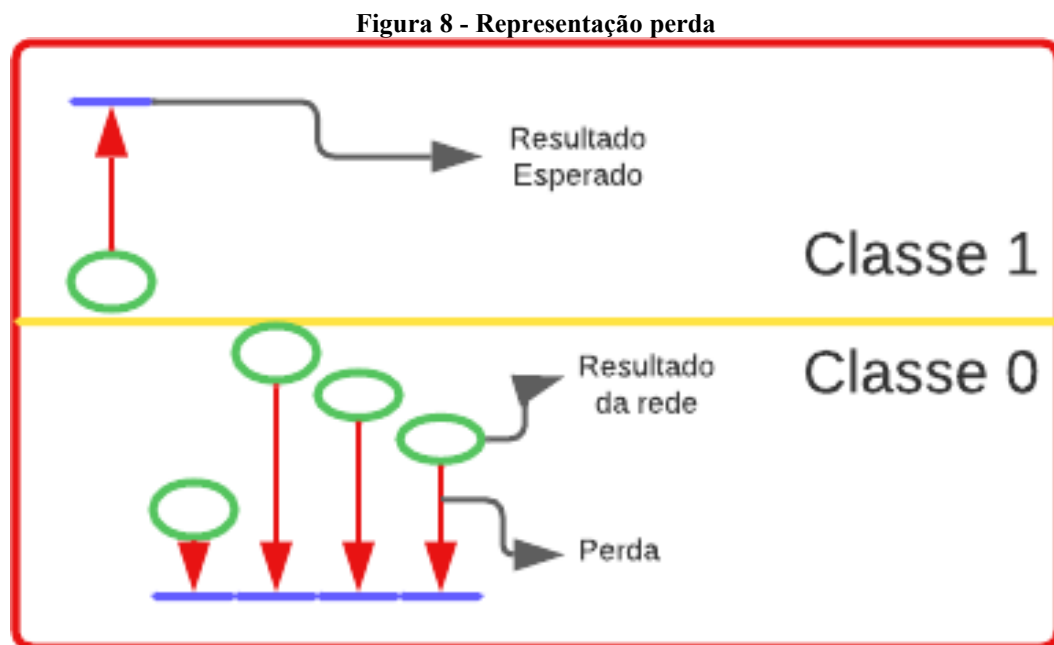
A avaliação do desempenho de uma rede neural é importante, tanto para validar a estrutura utilizada quando para analisar seus resultados. Existem vários parâmetros que podem ser observados para determinar o desempenho de uma rede neural (LIU; QI, 2017).

Para a aplicação proposta iremos analisar dois parâmetros de desempenho da rede neural, precisão que se refere a quantidade de classificações corretas que a rede fez, em relação a todas as feitas, podendo ser expressa matematicamente, conforme abaixo: (GOOGLE, 2022a)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (12)$$

Sendo que  $TP$  é a classificação correta de resultado positivo,  $TN$  é a classificação correta de resultado negativo,  $FP$  é a classificação errada de resultado positivo, e  $FN$  é a classificação errada de resultado negativo.

O outro parâmetro a ser analisado é a perda, esta função mede a diferença entre o resultado esperado e o entregue pela rede neural. Esta função é utilizada para chegar nos gradientes que por sua vez são usados para atualizar os pesos sinápticos de cada neurônio, na Figura 8 está representado o significado da perda (GOOGLE, 2022a; SARAVANAN, 2021; SEB, 2021).



Fonte: Autoria própria (2022)

Como pode ser observado na Figura 8, apesar da rede neural apresentar precisão máximo, pois acertou em todas suas classificações, a perda está alta, fazendo com que a confiabilidade dos resultados seja baixa. Um exemplo desta ocorrência são classificações em que o resultado da rede está próximo dos 50%, pois mesmo que o resultado esteja dentro da classificação esperada, a confiabilidade desta resposta não é ideal.

A perda é expressa matematicamente de diversas maneiras, a depender da aplicação e estrutura da rede neural. Para a utilização proposta por este trabalho, que é o desenvolvimento de uma rede classificadora binária, a função que mais se adequa é a entropia binária cruzada, definida da seguinte maneira:

$$Loss = -\frac{1}{N} * \sum_{i=1}^N (y_i * \log(\check{y}_i) + (1 - y_i) * \log(1 - \check{y}_i)) \quad (13)$$

Sendo que  $N$  é o número total de pontos analisados,  $y_i$  é o resultado esperado, e  $\check{y}_i$  é o resultado retirado da rede neural. Como o resultado esperado é sempre uma grandeza binária,



um dos termos irá ser zero, portanto, a equação 13 pode ainda ser rescritas em suas formas específicas.

$$Loss_1 = -\frac{1}{N} * \sum_{i=1}^N (\log(\check{y}_i)) \quad (14)$$

$$Loss_0 = -\frac{1}{N} * \sum_{i=1}^N (\log(1 - \check{y}_i)) \quad (15)$$

Sendo que as equações 14 e 15 representam a função perda quando a resposta esperada é um e zero respectivamente. Com isso fica claro que a perda é a média do logaritmo das distâncias entre os resultados obtidos e os esperados.

### 3 SIMULAÇÕES DE FALTAS PARA CRIAÇÃO DO BANCO DE DADOS

Para a criação do banco de dados foi utilizado um modelo computacional do motor de indução trifásico. Este modelo utiliza a modelagem dinâmica do motor de indução no plano ortogonal conforme desenvolvido por (VAS, 1998), além disso faz uso da função do MatLab ODE 45 para solução das equações diferenciais.

Optou-se pela utilização de simulações devido à indisponibilidade de máquinas com avarias para realização de testes e do elevado custo associado à criação deste banco de dados de forma física. O funcionamento do motor foi simulado em diversas condições de carga, e de intensidade de falta, criando-se dados suficientes para que o RNA desenvolvido tenha uma acuracidade adequada.

Para simular as faltas, que foram curtos-circuitos na bobina do estator, com diferentes intensidades, a resistência e a indutância do estator foram variadas em diferentes porcentagens de seu valor nominal, conforme equacionado na seção 2.2.1. Para a utilização no banco de dados, os períodos transitórios no qual o motor está sujeito devem ser ignorados, ou seja, apenas dados referentes ao estado de regime permanente do motor foram adicionados ao banco de dados, isto é feito pois caso os períodos transitórios fossem usados para o treinamento do RNA, a acurácia da rede iria diminuir. Além disso, para melhor representar a utilização da tecnologia proposta em meio industrial, a cada 5 segundos de simulação, aproximadamente 160 *ms* de dados estão sendo armazenados, considerado ainda que esses dados estão sendo amostrados do resultado dado pela simulação, com uma frequência de aproximadamente 625 hertz, com isso a quantidade de dados a serem armazenados e processados diminui significativamente, representando assim um sistema de menor custo de implantação em uma planta real.

#### 3.1 Dados do Motor simulado

O motor simulado possui os valores nominais conforme exposto na Tabela 2.

**Tabela 2 - Dados nominais do motor simulado**

Grandeza	Valor	Unidade
Resistencia do estator	0,9174	( $\Omega$ )
Indutância própria do estator	0,005473	(H)
Indutância Mútua	0.1854	(H)
Resistencia do rotor	0,6258	( $\Omega$ )
Indutância do rotor	0,005473	(H)
Velocidade	1800	(RPM)
Tensão de alimentação	469,48	(V)

Fonte: Vas (1998)

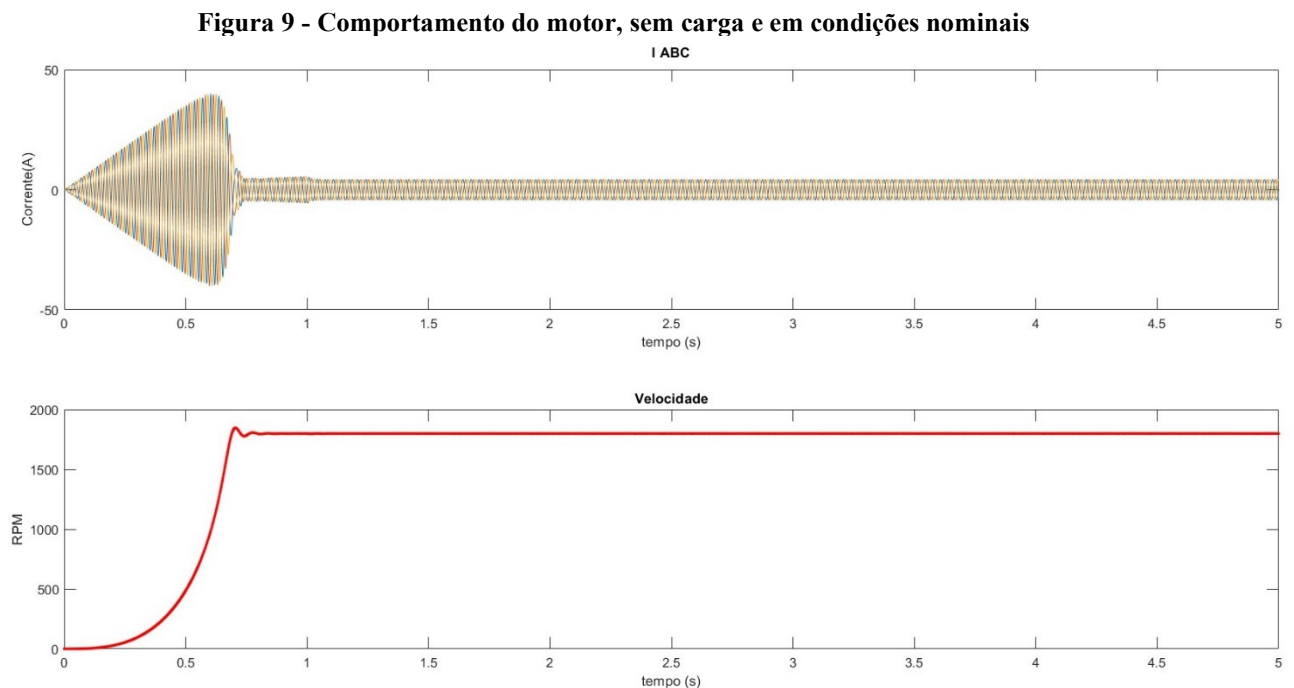
### 3.2 Resultados simulação

Para a RNA ser capaz de determinar corretamente que o motor está em seu funcionamento normal, foram feitas e adicionadas ao banco de dado várias simulações em que o motor funciona sem ocorrência de qualquer falta e com diferentes valores de carga.

Sendo que as simulações do motor foram feitas considerando uma partida em rampa de tensão, similar a uma partida suave de *soft-starters*, sendo esta uma prática comum para motores de maior potência.

#### 3.2.1 Simulação motor sem carga e sem falta

Inicialmente foi feita a simulação do motor, sem ocorrência de curto-circuito no estator e sem nenhuma carga. Por meio da Figura 9 é possível observar o comportamento da máquina, através das correntes de linha e da velocidade do motor.

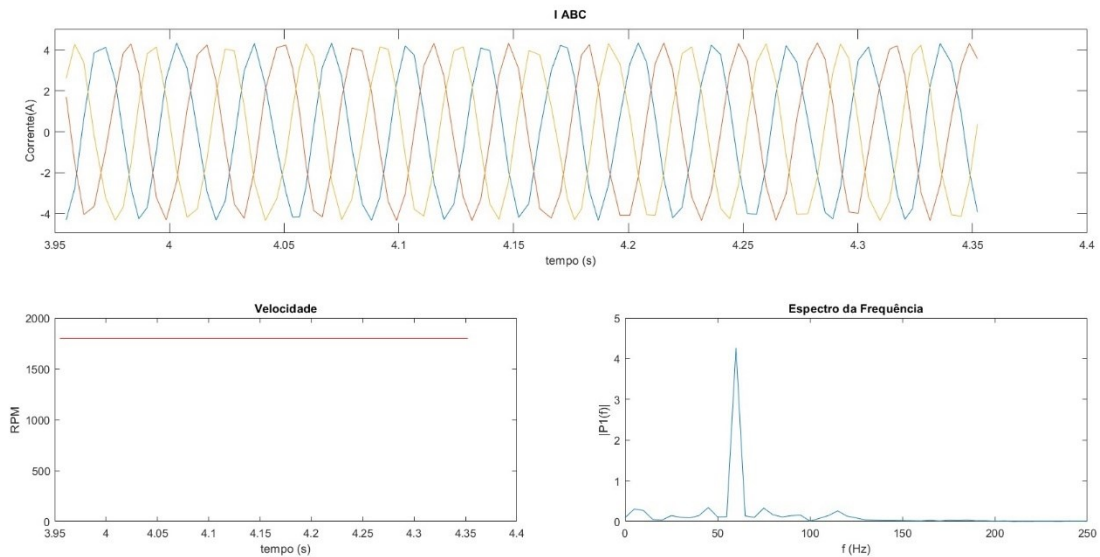


**Fonte: Autoria própria (2022)**

Analisando a figura acima, é possível verificar que após o período de partida, o motor manteve sua corrente e velocidade sem distúrbios, atingindo uma corrente máxima em regime de 4,35 Amperes, e uma velocidade de 1799 rotações por minuto. Mas durante a partida, como se era esperado, tanto a corrente quanto a velocidade apresentam um período transitório, com a corrente chegando a atingir valores de até 39 Amperes. Conforme discutido neste capítulo, os períodos transitórios da simulação foram desconsiderados, e uma seção de aproximadamente

160 ms foi amostrada para ser utilizada no treinamento da Rede Neural na Figura 10 é exposto os resultados dessa amostragem e da transformada rápida de Fourier da corrente de linha A.

**Figura 10 - Dados amostrados simulação sem carga e sem falta**



**Fonte: Autoria própria (2022)**

Como pode ser observado na imagem, o processo de amostragem, gera alguns desvios no formato das ondas de corrente, quando comparado ao resultado da simulação. No entanto a rede neural deve estar preparada para lidar com essas características, simulando assim de forma mais precisa o funcionamento do sistema em meio comercial.

Na Figura 10 está disposta também a transformada rápida de Fourier do sinal de corrente amostrada, fica claro que o sinal de corrente analisado, em regime, possui pouca influência harmônica e sub-harmônica, sendo que conforme descrito acima é necessário ainda levar em consideração a distorção da corrente devido ao processo de amostragem, que é refletido na transformada rápida de Fourier como um aumento na amplitude das componentes distintas da frequência fundamental.

Por fim, na Figura 10 pode ser visto também a velocidade amostrada do motor em regime, diferentemente da corrente, a velocidade é uma grandeza mais estável, possui pouca variação, sendo assim o processo de amostragem não prejudica sua representação.

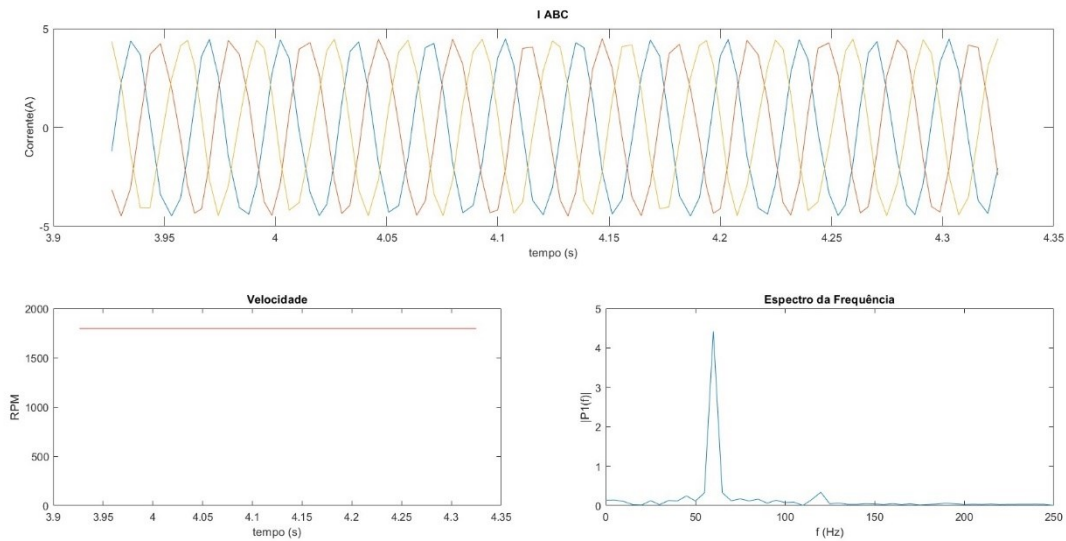
### 3.2.2 Simulação do motor sem ocorrência de faltas e com carga

Ao aumentar a carga em um motor de indução, conseqüentemente a corrente no seu estator também irá aumentar. O aumento de corrente sem que haja nenhuma variação na tensão, é um efeito similar aquele gerado por curtos no estator da máquina, portanto é necessário

alimentar o banco de dados, com várias combinações de carga e falta, para que a rede neural seja capaz de indicar com segurança a ocorrência de faltas, não percebendo aumentos na corrente causados por variações na carga como faltas.

Na Figura 11 estão expostos os resultados das grandezas amostradas, para a simulação do motor sem ocorrência de falta, com uma carga de 5 Nm.

**Figura 11 - Dados amostrados simulação carga 5 Nm e sem falta**



**Fonte: Autoria própria (2022)**

Conforme esperado, ao aumentar a carga, o motor reagiu com um aumento em sua corrente que em regime passou a atingir um valor máximo de 4,485 Amperes, e uma diminuição da velocidade que se estabilizou em aproximadamente 1795 rotações por minuto. A transformada rápida de Fourier apresentou resultado equivalente ao motor sem carga.

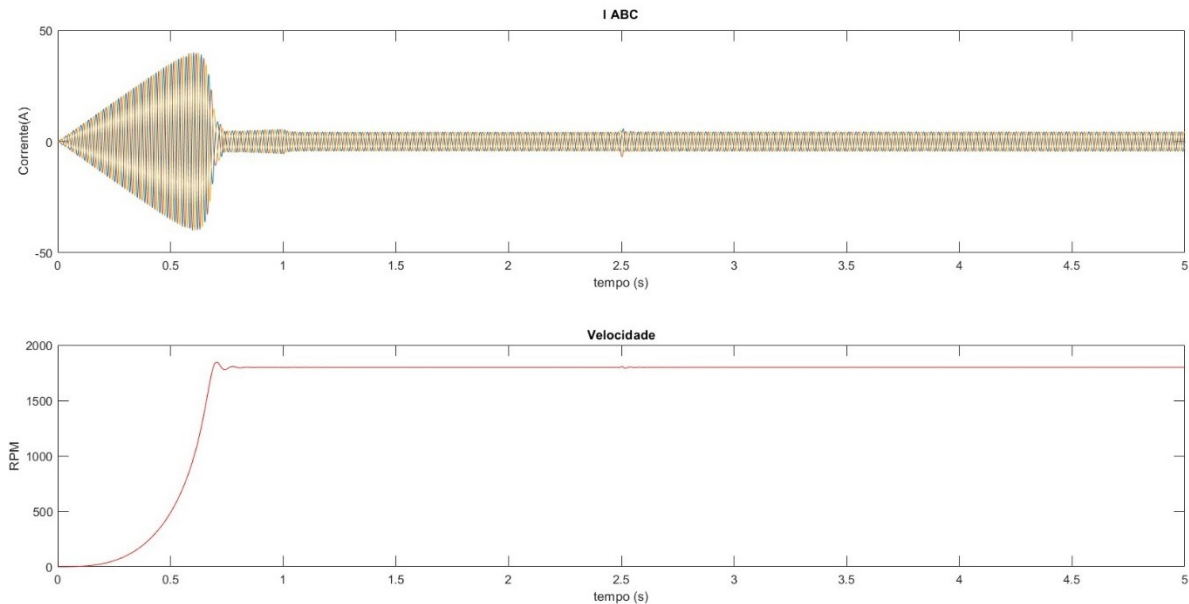
### 3.2.3 Simulação do Motor Sem carga e Com Ocorrência de faltas

Nesta seção será discutido os resultados obtidos simulando um motor que apresenta curto no estator, mas não possui carga. É válido ressaltar que situações similares a esta em meio industrial são incomuns, pois não é corriqueiro que motores fiquem atuando em regime sem a presença de carga, mas para testar e demonstrar a capacidade da rede neural os dados adquiridos nesta simulação foram adicionados ao banco de dados, e usados para o treinamento da rede neural.

De forma análoga a seção anterior, primeiramente será exposto e analisado os resultados obtidos na simulação, após isso, o mesmo será feito para os resultados amostrados. A Figura 12 mostra os resultados obtidos para a simulação, onde o motor opera descarregado e

aos 2,5 segundos ocorre uma falta com intensidade de 4%, ou seja, 4% de todas as espiras do estator entraram em curto-circuito.

**Figura 12 - Comportamento do motor, sem carga, com falta 4%**



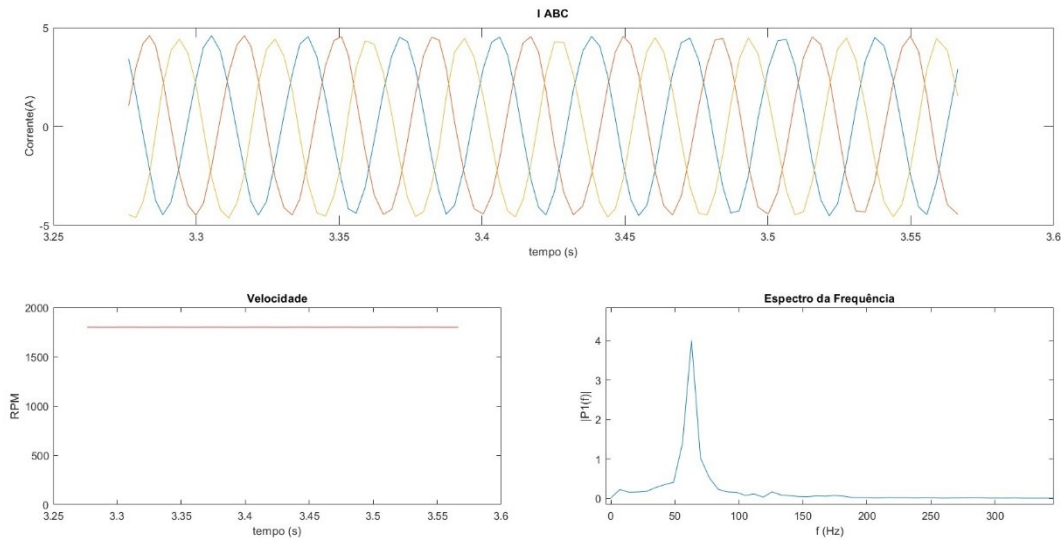
**Fonte: Aatoria própria (2022)**

Observando a Figura 12 fica claro a ocorrência de alguma condição extraordinária no motor, pois até a marca de 2,5 segundos, seu comportamento era idêntico aquele apresentando na Figura 9. No entanto é possível observar a ocorrência de um transitório tanto na corrente quanta na velocidade, sendo essa, a marca da introdução do curto-circuito nas bobinas do estator da máquina, após o transitório, conforme esperado houve um aumento tanto na corrente quanta na velocidade do motor. Estes aumentos nas grandezas são de uma escala tão baixa, que é extremamente difícil para as proteções convencionais dos ativos se sensibilizarem para elas, podendo até mesmo estar dentro da faixa de erro prevista para os equipamentos de medição.

Após o curto-circuito a corrente da máquina aumentou para um valor máximo de 4,49 amperes, um aumento relativo de 3%, em relação ao valor de regime permanente de pré-falta que foi de 4,35 amperes, já a velocidade apresentou um aumento relativo de 0,002%. Em meio industrial, onde uma série de fatores externos podem influenciar no funcionamento da máquina, como por exemplo: queda na tensão de alimentação, variação na frequência da rede, variações na carga, variações na temperatura ambiente, entre outros, a identificação dessas variações pode não ser trivial, dificultando assim a detecção da falta.

Na Figura 13 está representado os dados amostrados das curvas apresentadas na Figura 12.

**Figura 13 - Dados amostrados simulação sem carga, falta 4%**



**Fonte: Autoria própria (2022)**

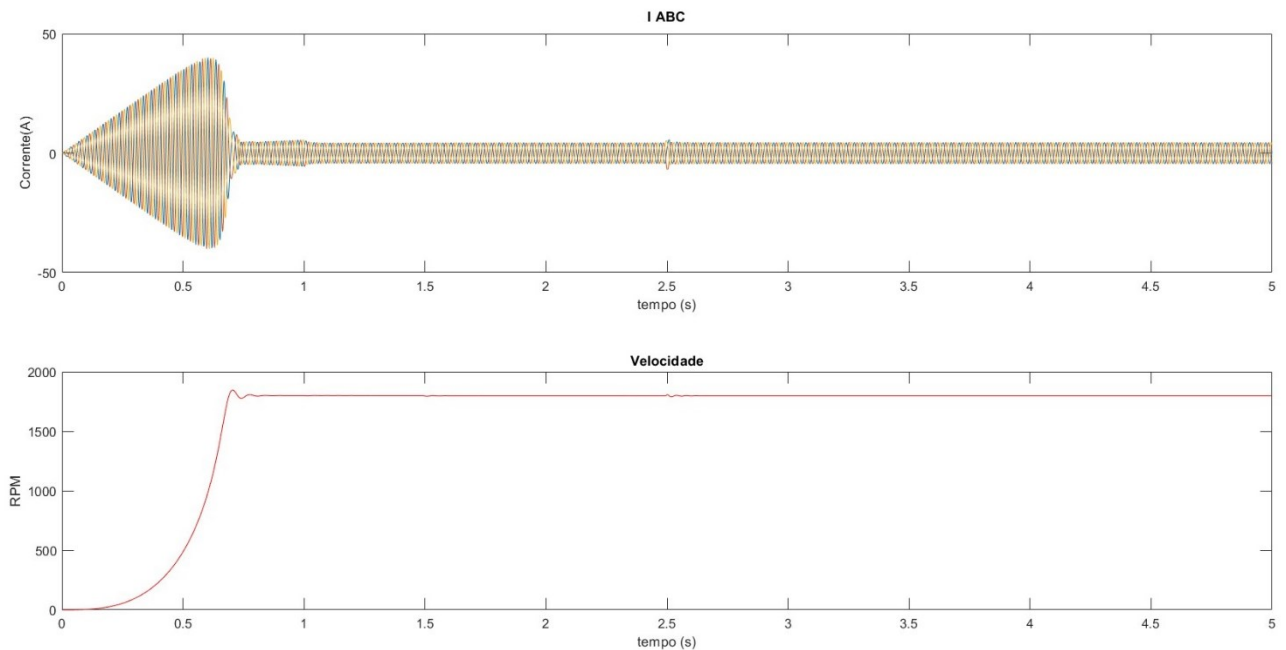
Comparando o resultado obtido da FFT exposta na Figura 13 com o representado pela Figura 10, aquilo que mais se destaca é a maior quantidade de componentes de maior frequência na corrente do estator sobre falta. Este aumento, prova o valor da análise do FFT, pois é um parâmetro que ajuda a rede neural identificar os padrões de comportamento do motor sobre curto-circuito nas espiras do estator.

### 3.2.4 Simulação do Motor Com carga e Com Ocorrência de faltas

Nesta seção será exposto e discutidos os resultados da simulação quando ocorre um curto-circuito no estator, quando o motor está sob carga. Esta condição é de grande importância, pois é aquela que mais se aproxima da aplicação real da detecção de faltas no estator.

Na Figura 14 pode ser observado os resultados obtidos na simulação, sendo que o motor partiu sem carga, aos 1,5 segundos foi adicionado a ele uma carga de 3 newtons-metro, e pôr fim aos 2,5 segundos foi introduzido um curto-circuito com intensidade de 2% ao seu estator.

**Figura 14 - Comportamento do motor, carga 3 Nm, falta 2%**



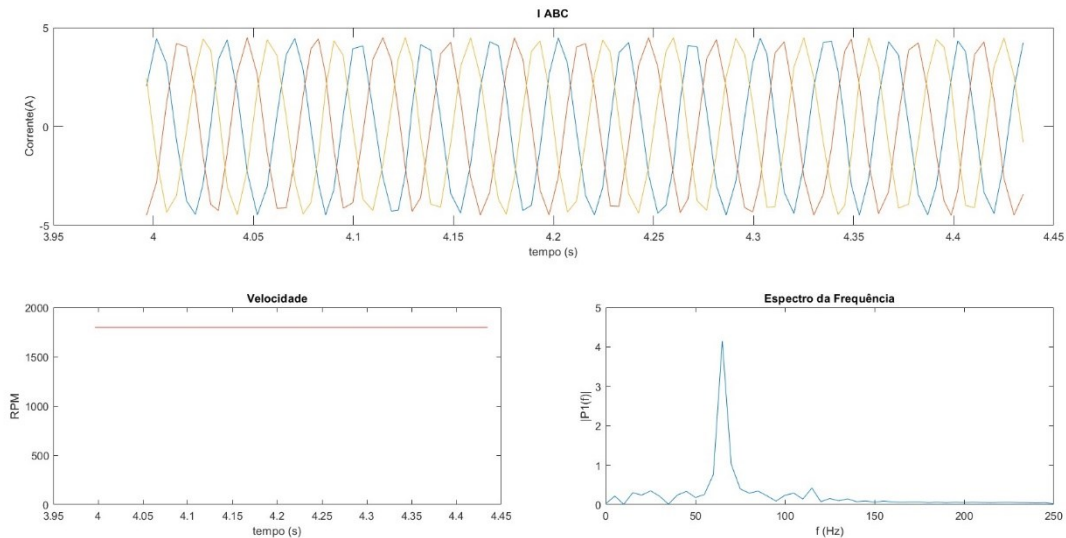
**Fonte: Autoria própria (2022)**

Inicialmente o motor funcionou normalmente, conforme discutido anteriormente na análise da Figura 9. Na marca de 1,5 segundos com a introdução da carga, houve um pequeno transitório na corrente e na velocidade, após isto o motor entrou em regime novamente, atingindo uma corrente máxima de 4,40 amperes e uma velocidade de 1797 rotações por minuto. Em 2,5 segundos a falta foi introduzida, fazendo com que haja um transitório no motor maior do que aquele ocorrido devido ao chaveamento da carga, a corrente em regime pós-falta atingiu valores de 4,497 amperes um aumento de aproximadamente 2% no valor da corrente, e um aumento de 0.060% rotações por minuto.

Analisando a Figura 14 é possível observar a semelhança do comportamento da máquina para uma falta e um chaveamento de carga, principalmente em relação a amplitude da corrente. Com isso fica claro que o comportamento da máquina perante um aumento de carga e a ocorrência de um curto-circuito em seu estator é dificilmente diferenciável.



**Figura 15 - Dados amostrados simulação carga 3 Nm, falta 2%**



**Fonte: Autoria própria (2022)**

Na Figura 15 está sendo mostrado os dados amostrados referentes as curvas da Figura 14. Como se era esperado a FFT da corrente apresentou uma maior intensidade nas componentes de alta frequência, mas ainda assim com magnitudes menores aquelas na simulação de falta com intensidade de 4%, portanto é possível concluir que o aumento da intensidade dos termos de alta frequência varia conforme a intensidade de falta.

### 3.3 Banco de dados

Para treinamento e teste da rede neural, é necessário a utilização de um banco de dados, contendo as informações de funcionamento da máquina em diversas condições. Conforme discutido em seções anteriores, os dados foram coletados através de simulações do funcionamento da máquina, realizou-se um total de 56 simulações, divididas entre sete diferentes intensidades de falta e oito níveis de carga, para cada simulação eram adicionadas ao banco de dados 7 parâmetros, com cem pontos cada.

Para um bom desempenho da rede neural, após a coleta dos dados é necessário seu tratamento. O primeiro processo realizado, foi a amostragem dos dados adquiridos, pois isto diminui significativamente a quantidade de dados a serem analisados pela rede, consequentemente diminuindo o esforço computacional para sua utilização. Além disto os dados também foram truncados, limitando-os a quatro centésimos, com isso diminuindo ainda mais a capacidade de processamento necessária.

Como última medida de tratamento, foi alterado o formato em que os dados são apresentados a rede neural. Devido a características do software utilizado, as informações

salvas, possuem o formato matricial de  $5600 \times 7$ , para que a rede neural detecte corretamente a falta, é necessário que todos os dados da simulação sejam *inputs* simultaneamente, com isso o formato dos dados foi alterado para  $700 \times 56$ , fazendo todas as informações relacionadas a uma simulação esteja em apenas uma linha da matriz.

Para o processo de treinamento é necessário também, indicar quais são as saídas relacionadas a cada entrada da rede neural. Portanto foi desenvolvida separadamente uma matriz de saídas, com o formato  $56 \times 1$ , onde 8 pontos desta matriz representavam o motor em funcionamento normal, e o restante indicavam faltas.

#### 4 DESENVOLVIMENTO DO SISTEMA INTELIGENTE

Para o desenvolvimento do sistema inteligente aplicando rede neural artificial, foi utilizado a linguagem de programação Python com a biblioteca TensorFlow. Foi optado por utilizar estas ferramentas devido a ambas serem de código aberto, ou seja, podem ser utilizados gratuitamente, além disto esta biblioteca possui várias funções nativas que facilitam a criação e treinamento da rede neural (GOOGLE, 2022b)(HOLLOWAY, 2017).

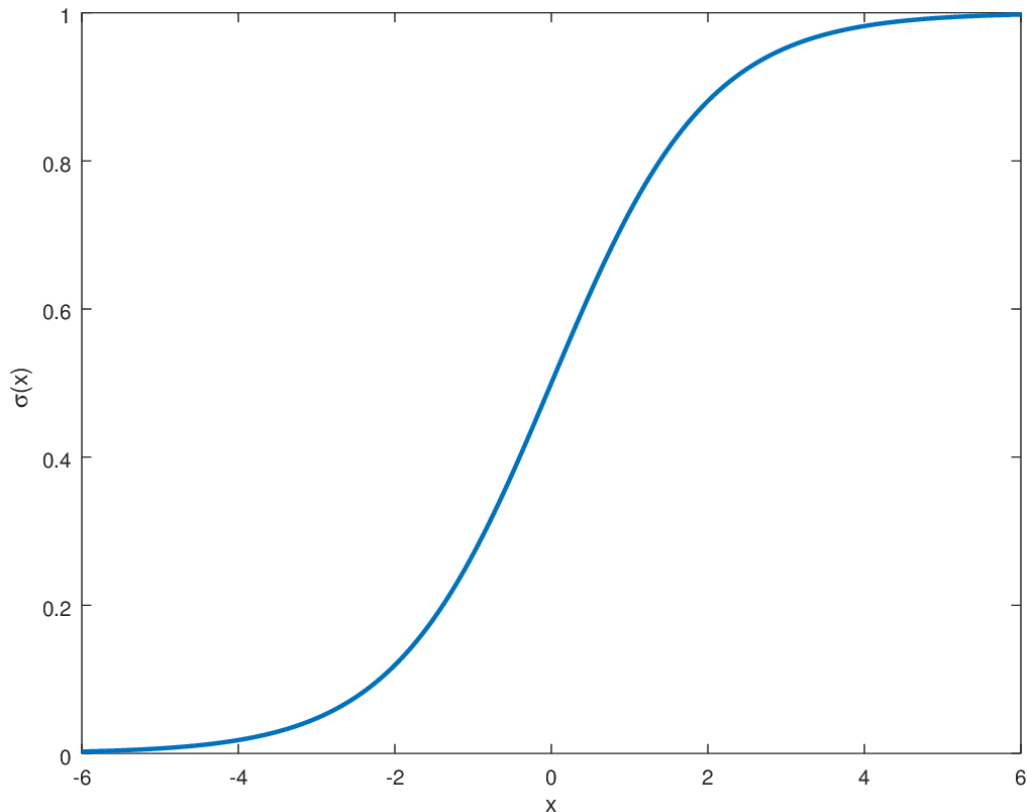
Os dados de entrada da rede serão os seguintes valores instantâneos: as tensões de fase ( $V_a, V_b, V_c$ ), correntes de linha ( $I_a, I_b, I_c$ ) e a transformada rápida de Fourier da corrente ( $FFT_{I_a}$ ), já sua saída é o estado da máquina (com falta = 1, sem falta = 0). O processo de treinamento foi feito utilizando os dados adquiridos durante as simulações computacionais, de modo que o programa seja capaz de reconhecer os padrões que são apresentados quando o motor está em falta.

O método de treinamento utilizado foi o aprendizado supervisionado, este método foi escolhido pois é aquele que possui melhor aplicabilidade no mercado, devido a ser possível formar bancos de dados individuais para cada máquina com os parâmetros medidos pelo seu monitoramento.

Como função de ativação, foi escolhido a utilização da função sigmoide, sendo essa uma função amplamente utilizada em *machine learning*, especialmente em casos em que se deseja ter uma conclusão que depende da probabilidade da ocorrência de algum fato, possui a seguinte definição matemática e representação gráfica: (SILVA; FREITAS; BISPO, 2019)

$$S(x) = \frac{1}{1 + e^{-x}} \quad (16)$$

Figura 16 - Gráfico sigmoide



Fonte: Berg (2018)

#### 4.1 Estrutura da rede neural

A rede neural escolhida consiste em quatro camadas, sendo a primeira a camada de entrada com 700 neurônios, a segunda e terceira camadas são ocultas, com 512 e 256 neurônios cada respectivamente pôr fim a quarta é a camada de saída com dois neurônios. Vários layouts de RNA foram testados, e essa foi a que apresentou a melhor eficiência em relação a sua acuracidade e esforço computacional.

No entanto, não é possível definir de forma concreta que a estrutura escolhida ou qualquer outra é intrinsecamente superior, pois variáveis como capacidade computacional disponível, quantidade de máquinas monitoradas simultaneamente e a dificuldade de acesso as máquinas influenciam significativamente a estrutura que melhor irá se encaixar na aplicação. Portanto será testado também diferentes estruturas que podem ser mais vantajosas em determinadas aplicações, sendo elas:

- rede neural de 6 camadas, sendo 4 delas ocultas, com número de neurônios variando de 1028 até 128, as camadas de entrada e saída se mantem conforme a estrutura escolhida.

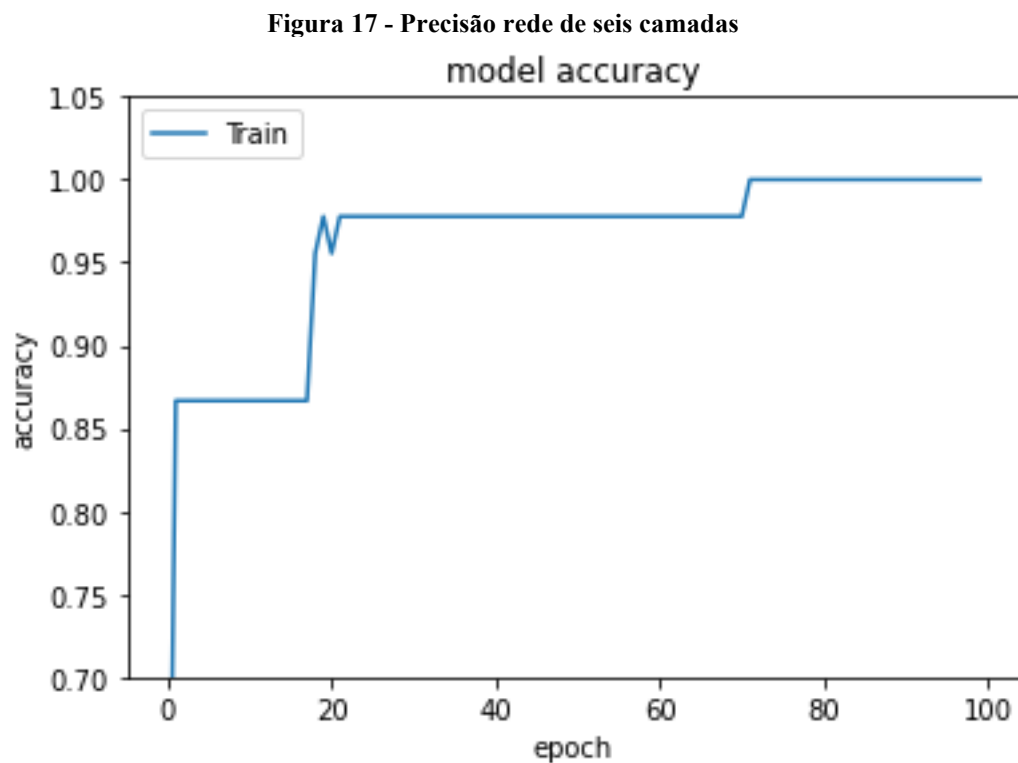
- rede neural de 3 camadas, com apenas uma camada oculta de 64 neurônios, as camadas de entrada e saída se mantem conforme a estrutura escolhida.

Outro parâmetro extremamente importante de se determinar para o treinamento das redes neurais é o número de épocas, este parâmetro representa o número de vezes que um mesmo dado irá passar pela rede. Podem ser encontradas diferentes maneiras em que as épocas são utilizadas dentro dos algoritmos de treinamento, como por exemplo, a definição de um valor de erro mínimo, e de um número máximo de épocas, neste caso a rede iria ser treinada até alcançar um valor abaixo do erro definido, ou então chegar ao número máximo de épocas permitido, no entanto na função utilizada para treinamento, é definido diretamente o número de épocas que deverá ser atingido.

Com isso fica claro que dependendo do valor escolhido para as épocas, a rede pode se tornar mais precisa, mas ao custo de um maior custo de processamento. A rede com maior número de camadas será testada utilizando as épocas em 100, a rede escolhida terá *épocas* em 25, por fim a rede com menor número de camadas utilizara *épocas* em 5.

#### 4.1.1 Rede de seis camadas

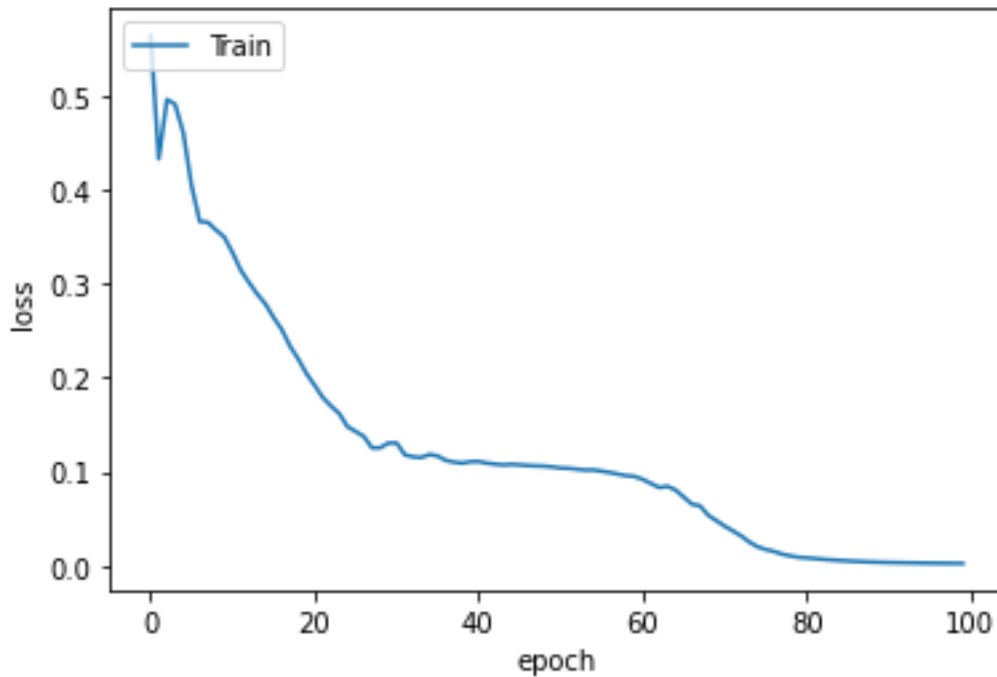
A rede de seis camadas foi treinada, e apresentou resultado de *accuracy* conforme Figura 17.



Fonte: Autoria própria (2022)

Analisando a figura acima, é possível observar que a rede chegou a uma precisão de 100%, ou seja, classificou corretamente todos os dados. Fica claro também que para o parâmetro analisado, a estrutura está sobredimensionada, atingindo precisão máxima ainda na época 73.

**Figura 18 - Perda rede de seis camadas**  
model loss



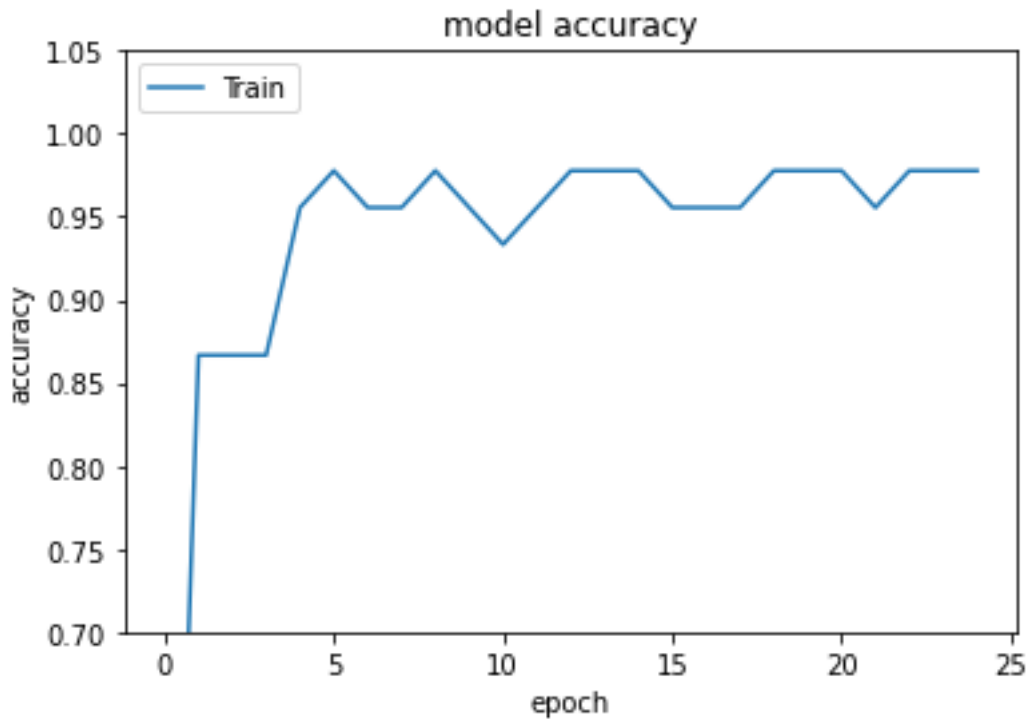
Fonte: Autoria própria (2022)

Conforme esperado é possível observar que a perda do sistema cai significativamente, conforme o treino progride, no final do treinamento esta rede apresentou uma perda de 0,00225, portanto é possível concluir que esta rede apresentou tanto uma ótima precisão em suas classificações, como também um grande grau de certeza em tais. No entanto, apresentou um tempo de processamento significativo, de aproximadamente 6 segundo, sendo que este tempo pode aumentar bruscamente, dependendo da máquina utilizada, ou da quantidade de dados.

#### 4.1.2 Rede de quatro camadas

De forma equivalente a rede de seis camadas, a rede de quatro camadas foi treinada e sua precisão está estrutura está representada pela Figura 19.

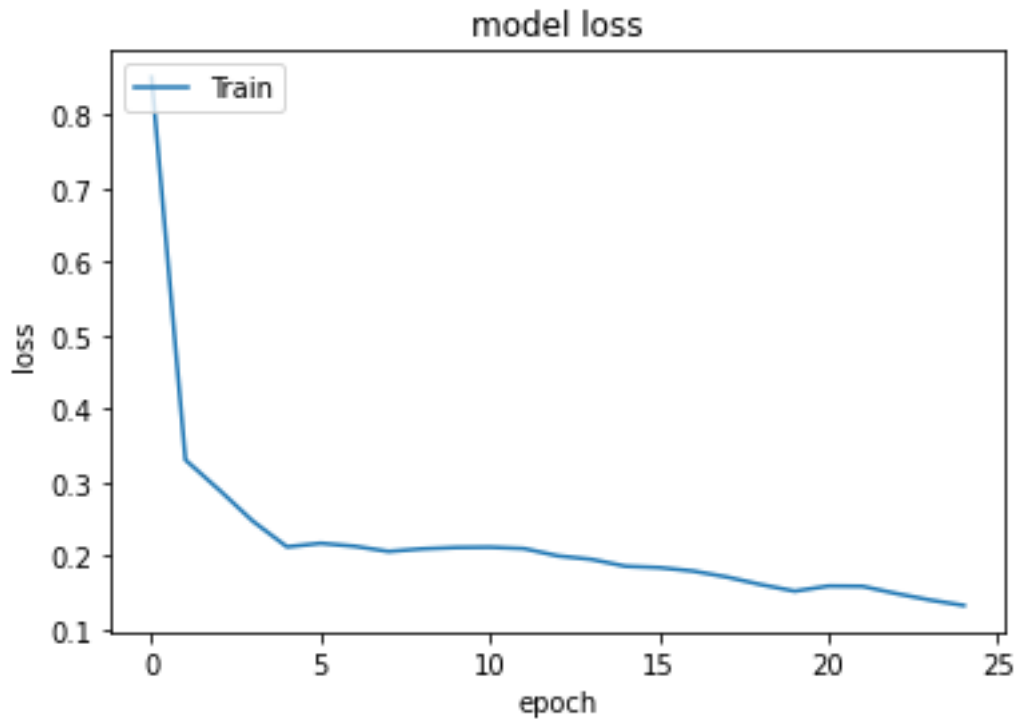
Figura 19 - Precisão rede de quatro camadas



Fonte: Autoria própria (2022)

É possível verificar que apesar da grande diminuição do número de neurônios e de épocas a precisão da rede não abaixou bruscamente. A precisão do RNA se estabeleceu em 0.9778, ou seja, considerando que o resultado do treinamento da rede reflete de forma ideal seu comportamento para outras entradas, a cada cem classificações a rede neural erraria apenas 2. Como pode ser visto na Figura 20, a perda desta rede se estabeleceu em 0.1325, um valor muito maior do que aquele observado na rede de seis camadas. Em contrapartida, o tempo de processamento requisitado para o treinamento desta rede foi de 2 segundos.

Figura 20 - Perda rede de quatro camadas

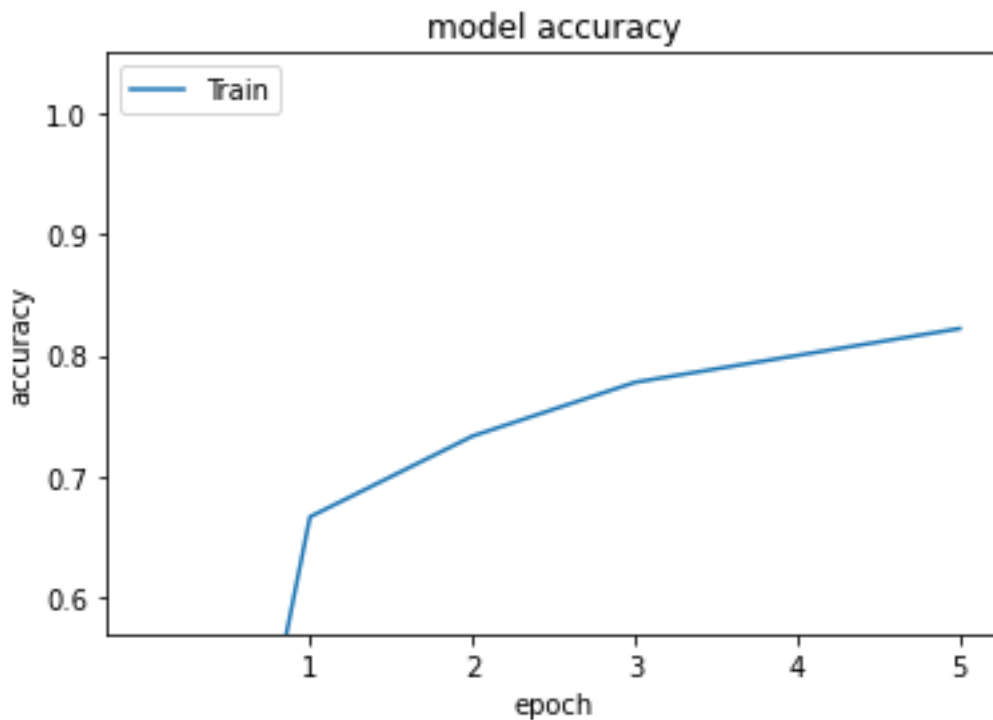


Fonte: Autoria própria (2022)

#### 4.1.3 Rede de três camadas

A rede neural com estrutura de três camadas foi treinada, e o resultado de precisão foi conforme Figura 21.

Figura 21 - Precisão rede de três camadas

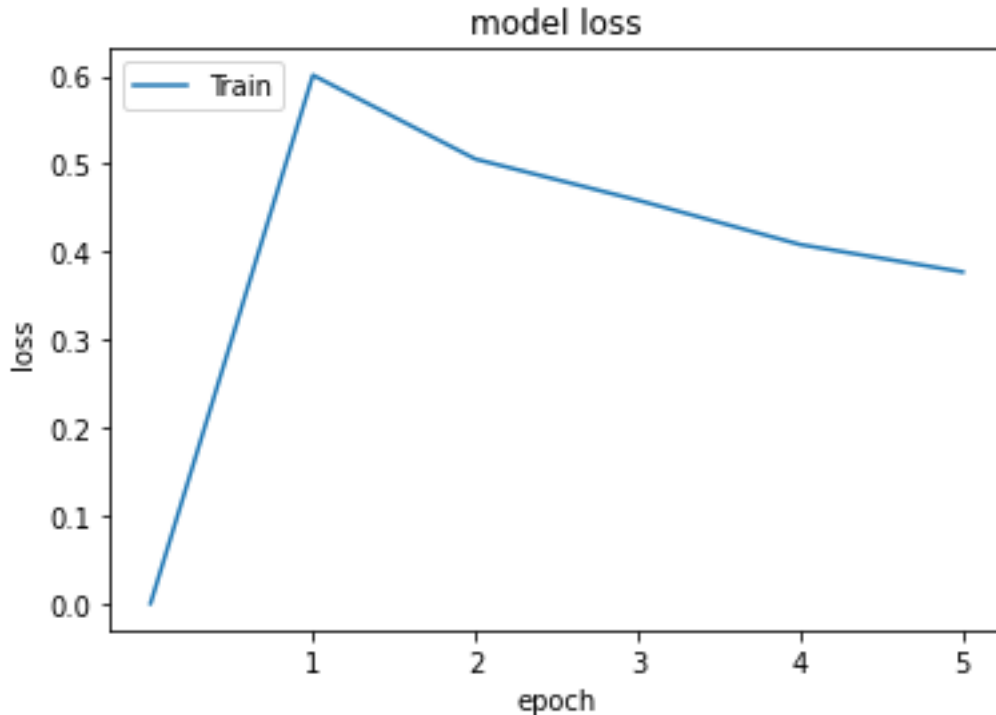


Fonte: Autoria própria (2022)



Ao final de seu treinamento, a rede de estabeleceu em uma precisão de aproximadamente 0.8222, conforme esperado, o menor valor entre as três estruturas.

**Figura 22 - Perda Rede de Três Camadas**



Fonte: Autoria própria (2022)

Observando a imagem acima, fica claro o ponto mais crítico nesta estrutura de rede, a perda em um valor muito elevado. Conforme citado anteriormente, esta alta perda gera uma grande incerteza sobre as classificações da rede, o que pode dificultar a tomada de ações, principalmente em ativos críticos ou de difícil acesso. Em contrapartida seu tempo treinamento foi de aproximadamente 0,4 segundos, cerca de 15 vezes menor do que o tempo da rede de seis camadas. Portanto, em aplicações em que existem muitas máquinas a serem monitoradas, e uma baixa capacidade de processamento, este formato de rede pode ser a mais aplicável.

#### 4.1.4 Comparação de classificação das estruturas

Para comparar as classificações feitas por cada uma das redes, foram escolhidos três pontos aleatórios do banco de dados de teste e cada uma das três redes classificou tais pontos.

O primeiro ponto representa o funcionamento do motor sem falta e sem carga o resultado obtido está exposto na Tabela 3.

**Tabela 3 - Resultado das estruturas para o ponto de teste 1**

Estrutura	Sem ocorrência de falta	Em falta
Rede de seis camadas	99,775 %	0,224 %
Rede de quatro camadas	58,281 %	41,719 %
Rede de três camadas	65,45 %	34,55 %

**Fonte: Autoria própria (2022)**

Analisando a Tabela 3 fica claro a diferença de certeza que cada estrutura tem sobre sua classificação. Conforme esperado, a rede de seis camadas gerou o melhor resultado, apresentando aproximadamente 100% de certeza de que não ocorreu nenhuma falta no motor, já a rede de quatro camadas teve cerca de 58% de certeza, valor menor até mesmo do que a rede de três camadas, que teve 65,45%. Apesar de todas as redes indicarem corretamente a não ocorrência de falta, a rede três e principalmente a quatro, ficaram próximos do limiar de 50%, portanto, para classificação deste ponto seria interessante aumentar o número de neurônios destas redes, ou aumentar o banco de dados de treinamento.

O segundo ponto testado, representa o funcionamento do motor em falta, com intensidade de 3% e uma carga de 7 newtons-metro, o resultado da classificação das redes pode ser visto na Tabela 4.

**Tabela 4 - Resultado das estruturas para o ponto de teste 2**

Estrutura	Sem ocorrência de falta	Em falta
Rede de seis camadas	0,213 %	99,787 %
Rede de quatro camadas	7,022 %	92,978 %
Rede de três camadas	50,499 %	49,501 %

**Fonte: Autoria própria (2022)**

A rede de seis camadas, manteve seu ótimo resultado, novamente apresentando uma certeza próxima dos 100% em sua classificação. Já a rede de quatro camadas, apresentou um resultado muito superior quando comparado ao do ponto um, pôr fim a rede de três camadas classificou este ponto de forma errônea, indicando que o motor nestas condições não estaria em falta, além disso observando a certeza da classificação, fica claro que este ponto gerou uma grande incerteza na rede, sendo que este resultado seria de pouca valia, mesmo que a classificação estivesse correta.

O último ponto testado, representa o motor em falta de intensidade 4% e carga de 6 Newtons metro, a classificação de cada rede, pode ser vista na Tabela 5.

**Tabela 5 - Resultado das estruturas para o ponto de teste 3**

Estrutura	Sem ocorrência de falta	Em falta
Rede de seis camadas	0,196 %	99,804 %
Rede de quatro camadas	0,440 %	99,560 %
Rede de três camadas	10,142 %	89,858 %

**Fonte: Autoria própria (2022)**

Por fim, neste ponto todas as redes apresentaram resultado satisfatório, sendo que a diferença entre as redes de quatro e seis camadas, é negligenciável, e até mesmo a rede mais simples apresentou uma certeza próxima dos 90%.

Com isso, fica claro a superioridade de desempenho da rede de seis camadas, no entanto, em muitos pontos sua diferença em relação a rede de quatro camadas é mínima, não justificando o aumento na necessidade de processamento. Já a rede de três camadas apresentou o pior desempenho, sofrendo demasiadamente devido a sua alta taxa de perda, fazendo com que seus resultados sejam pouco confiáveis.

#### **4.2 Aplicação e constante desenvolvimento da rede neural**

Para que a rede neural, forneça e se mantenha a fornecer um resultado confiável, é necessário que a rede seja constantemente treinada e tenha seu banco de dados atualizados com os novos dados adquiridos. Devido ao desgaste natural sofrido pelas máquinas, dados coletados anteriormente podem não mais representar de forma correta o comportamento do motor, portanto é recomendado que novos dados sejam coletados e substituam os antigos no banco de dados.

Além disso é interessante que se haja um período de teste e observação da máquina, verificando as diferentes condições em que a mesma trabalha, e qual a resposta da rede perante elas. Também é possível utilizar outras entradas que tem influência no comportamento da máquina, assim melhorando a precisão da rede neural, por exemplo: tensão, frequência e temperatura.

## 5 CONCLUSÃO

Através do desenvolvimento deste trabalho, foi possível observar como a implementação da indústria 4.0 pode ser vantajosa, tanto em um aspecto de diminuição dos custos de produção, quanto para o aumento da produtividade e eficiência dentro das indústrias. Sendo que o monitoramento das máquinas para detecção de faltas apresenta grandes benefícios, considerando o grande impacto que a falha em uma máquina pode causar.

Foi utilizado um modelo computacional para simulação de faltas no estator de um motor elétrico de indução, com isso foi verificado como um curto-circuito nas bobinas do estator pode influenciar o funcionamento da máquina. Através dos dados adquiridos neste modelo, foi feito o treinamento de diversas estruturas de redes neurais, onde se foi discutido diferentes situações em que cada uma dessas estruturas poderia ser aplicada, além de chegar um uma estrutura generalista, que pode ser aplicada na maioria dos casos.

No entanto é necessário notar que, as máquinas em uma linha de produção podem estar expostas as mais diversas variáveis, fazendo assim com que o treinamento do sistema inteligente seja um grande desafio. Pois uma máquina exposta a determinadas condições adversas pode não apresentar o mesmo comportamento de uma máquina exposta a condições ideais, portanto seria necessário para o melhor funcionamento do sistema, que os dados para treinamento sejam adquiridos na máquina a qual o sistema irá monitorar. Além disso uma alteração nas condições de funcionamento da máquina pode impactar na confiabilidade da análise, fazendo-se necessário que o sistema passe por um novo processo de treinamento, ou então que seja aplicado técnicas de *Machine Learning* que aplicam treinamento contínuo no sistema, sendo esses grandes desafios para sua implementação.

## REFERÊNCIAS

ANTONIO, M.; LOSADA, M. Influência de Fatores de Manutenção que Afetam a Vida Útil dos Motores de Indução. *In: VII Conferência Internacial de Aplicações Industriais*. 10., 2008, Poço das Caldas. **Anais [...]** Manhattan: IEEE, 2008, p. 1–7.

BAZAN, G. H. **Medidas de informação e sistemas inteligentes aplicados no diagnóstico de curto-circuito do estator de motores de indução trifásicos**. Dissertação (mestrado em engenharia elétrica) – Universidade Tecnológica Federal do Paraná – Programa de Pós-Graduação em Engenharia Elétrica, Cornélio Procópio, 2016.

BERG, H. **The sigmoid function (a.k.a. the logistic function) and its derivative**.

Disponível em: <[https://hvidberrrg.github.io/deep\\_learning/activation\\_functions](https://hvidberrrg.github.io/deep_learning/activation_functions)>. Acesso em: 15 jun. 2022

CHAPMAN, S. J. **Fundamentos de Máquinas Elétricas**. Porto Alegre: AMGH, 2013

CHOUDHARY, A. *et al.* Condition Monitoring and Fault Diagnosis of Induction Motors: A Review. **Archives of Computational Methods in Engineering**, v. 26, n. 4, p. 1221–1238, 2019.

DAVIES, R.; COOLE, T.; SMITH, A. Review of Socio-technical Considerations to Ensure Successful Implementation of Industry 4.0. **Procedia Manufacturing**, v. 11, n. June, p. 1288–1295, 2017.

DUAN, F.; ZIVANOVIC, R. A Model for Induction Motor with Stator Faults. *In: 2012 22nd Australasian Universities Power Engineering Conference (AUPEC)*. 9., 2012, Bali. **Anais[...]** Manhattan: IEEE, 2012. p. 1-5.

EFTEKHARI, M. et al. Online detection of induction motor’s stator winding short-circuit faults. **IEEE Systems Journal**, v. 8, n. 4, p. 1272–1282, 1 dez. 2014.

FREITAS, T. **Os três tipos de aprendizado no machine learning, um ramo da inteligência artificial**. Disponível em: <<https://www.startse.com/noticia/nova-economia/machine-learning-inteligencia-artificial-aprendizado>>. Acesso em: 14 ago. 2021.

GOLDMAN, F. L.; LAU, V. L.; MONTEIRO, L. D. S. Um modelo conceitual para a indústria 4.0: o atual desafio da engenharia de produção no Brasil. **Revista da jopic** p. 100–105, 2020.

GOOGLE. **Module: tf**. Disponível em: [https://www.tensorflow.org/api\\_docs/python/tf](https://www.tensorflow.org/api_docs/python/tf). Acesso em: 22 jan. 2022a.

GOOGLE. **Machine Learning Crash Course with TensorFlow APIs**. Disponível em: <https://developers.google.com/machine-learning/crash-course/classification/accuracy>. Acesso em: 23 set. 2022b.

HOLLOWAY, O. **From Solving Equations to Deep Learning: A TensorFlow Python Tutorial**. Disponível em: <https://www.toptal.com/machine-learning/tensorflow-python-tutorial>. Acesso em: 10 out. 2021

LAVVA, A. **Difference Between Single Phase and Three Phase Induction Motor: All you need to know**. Disponível em: <https://www.linquip.com/blog/single-phase-vs-three-phase-induction-motor/>. Acesso em: 9 ago. 2021.

LEE, J. *et al.* Industrial Artificial Intelligence for industry 4.0-based manufacturing systems. **Manufacturing Letters**, v. 18, p. 20–23, 2018.

LIU, L.; QI, H. Learning effective binary descriptors via cross entropy. *In*: 2017 IEEE Winter Conference on Applications of Computer Vision, 3., 2017, Santa Rosa. **Anais [...]** Manhattan: IEEE, 2017. p. 1251-1258.

MESQUITA, D. **Python AI: How to Build a Neural Network & Make Predictions**. Disponível em: < <https://realpython.com/python-ai-neural-network/> >. Acesso em: 2 ago. 2021

QI, Q.; TAO, F. Digital Twin and Big Data Towards Smart Manufacturing and Industry 4.0: 360 Degree Comparison. **IEEE Access**, v. 6, p. 3585–3593, 2018.

SANTOS, M.; MANHÃES, A. M.; LIMA, A. R. Indústria 4.0: desafios e oportunidades para o Brasil. São Cristóvão. *In*: SIMPÓSIO DE ENGENHARIA DE PRODUÇÃO DE SERGIPE, 10., 2018, São Cristóvão. **Anais [...]** São Cristóvão: UFS, 2018. P. 317-329.

SARAVANAN, P. **Understanding Loss Functions in Machine Learning**. Disponível em: <https://www.section.io/engineering-education/understanding-loss-functions-in-machine-learning/>. Acesso em: 20 ago. 2022

SEB. **An Introduction to Neural Network Loss Functions**. Disponível em: <https://programmatically.com/an-introduction-to-neural-network-loss-functions/#:~:text=The%20loss%20function%20in%20a,all%20losses%20constitutes%20the%20cost.>>. Acesso em: 23 set. 2022.

SHARIFI, R.; EBRAHIMI, M. Detection of stator winding faults in induction motors using three-phase current monitoring. **ISA Transactions**, v. 50, n. 1, p. 14–20, 2011.

SILVA, F.; FREITAS, P.; BISPO, S. **Inteligência Artificial**. Rio de Janeiro: 2019.

SILVA, I. N.; SPATTI, D. H.; FLAUZINO, R. A. **Redes Neurais Artificiais Para Engenharia e Ciências Aplicadas**. São Paulo, 2010.

SUETAKE, M. **Sistemas Inteligentes Para Monitoramento e Diagnósticos de Falhas em Motores de Indução Trifásicos**. 2012. Tese (doutorado em engenharia elétrica) – Universidade de São Paulo, Programa de Pós-Graduação em Engenharia Elétrica, São Carlos, 2012.

VAS, P. **Sensorless Vector and Direct Torque Control**. Oxford University Press, 1998.

VERMULM, R. Políticas para o Desenvolvimento da Indústria 4.0 no Brasil. **Instituto de Estudos para o Desenvolvimento Industrial**, p. 30, 2018.

VITOR, A. L. D. O. *et al.* Detecção de falhas elétricas em motores de indução utilizando rede radial basis function. *In: XX Congresso Brasileiro de Automática*. 9., 2014 Belo Horizonte. **Anais[...]** Belo Horizonte: UFMS, 2014. p. 3830–3837





## **APÊNDICE A - Código fonte para criação e treinamento da rede neural**

Para criação e treinamento da rede neural foi desenvolvido um código em linguagem de programação Python. Neste programa é definido o formato da rede neural e seus parâmetros de treinamento, além disto também é necessário que os dados para treinamento e validação estejam em formato .CSV para que sejam carregados pelo código e utilizados.

```
# -*- coding: utf-8 -*-
```

```
"""
```

```
Created on Wed Feb 7 17:35:31 2022
```

```
@author: luis_
```

```
"""
```

```
# -*- coding: utf-8 -*-
```

```
"""
```

```
Spyder Editor
```

```
"""
```

```
# TensorFlow
```

```
import tensorflow as tf
```

```
from tensorflow import keras
```

```
#
```

```
import matplotlib.pyplot as plt
```

```
# Helper libraries
```

```
import numpy as np
```

```
import pandas as pd
```

```
import time
```

```
training_data = pd.read_csv(r'', delimiter=',') ##Dados para treinamento
```

```

training_data.shape

training_data.columns = ['va','vb','vc','Ia','Ib','Ic','FFTIa'] ## Dados para Validação

testing_data = pd.read_csv(r", delimiter=",")

testing_data.shape

testing_data.columns = ['va','vb','vc','Ia','Ib','Ic','FFTIa']

y_train1 = pd.read_csv(r", delimiter=",") ## resultados treinamento

y_train1.columns = ['falta']

training_data = tf.reshape(training_data, shape=[-1,700]) # resultados validação

y_testing1 = pd.read_csv(r", delimiter=",")

y_testing1 .columns = ['falta']

testing_data = tf.reshape(testing_data, shape=[-1,700])

y_train = np.asarray(y_train1).astype('float32').reshape((-1,1))

y_testing = np.asarray(y_testing1).astype('float32').reshape((-1,1))

y_train.shape

st= time.time()

model = keras.Sequential([

    keras.layers.Dense(700, input_shape=(700,)), # input layer (1)

    keras.layers.Dense(, activation='sigmoid'), # hidden layer (2)

    keras.layers.Dense(1, activation='sigmoid') # output layer (3)

])

model.compile(optimizer='adam',

```

```
        loss='binary_crossentropy',

        metrics=['accuracy'])

history = model.fit(training_data,y_train, epochs=) # we pass the data, labels and epochs and
watch the magic!

et = time.time()

tempo_total= et-st

print('Execution time:', tempo_total, 'seconds')

test_loss, test_acc = model.evaluate(testing_data, y_testing, verbose=2)

teste=model.evaluate(testing_data, y_testing , verbose=2)

plt.plot(history.history['accuracy'])

plt.ylim(bottom=0.7, top=1.05)

plt.title('model accuracy')

plt.ylabel('accuracy')

plt.xlabel('epoch')

plt.legend(['Train'], loc='upper left')

plt.show()

plt.plot(history.history['loss'])

plt.title('model loss')

plt.ylabel('loss')

plt.xlabel('epoch')

plt.legend(['Train'], loc='upper left')

plt.show()
```

```
print('Test accuracy:', test_acc)

predictions = model.predict(testing_data)

i=0

predictions[10]

np.argmax(predictions[2])

model.save("")
```

Para criação e treinamento da rede neural basta definir a localização dos arquivos de treinamento e validação, definir o formato da rede seus parâmetros de treinamento e sua identificação para que ela possa ser salva.