



UNIVERSIDADE FEDERAL DE MATO GROSSO
FACULDADE DE ARQUITETURA, ENGENHARIA E TECNOLOGIA
DEPARTAMENTO DE ENGENHARIA ELÉTRICA

JOÃO VITOR FABRIS

**APLICAÇÃO DE REDES NEURAS ARTIFICIAIS À ANÁLISE DE ESTABILIDADE DE
TENSÃO NO MONITORAMENTO DE SISTEMAS ELÉTRICOS DE POTÊNCIA**

CUIABÁ – MT
DEZEMBRO, 2022

JOÃO VITOR FABRIS

**APLICAÇÃO DE REDES NEURAS ARTIFICIAIS À ANÁLISE DE ESTABILIDADE DE
TENSÃO NO MONITORAMENTO DE SISTEMAS ELÉTRICOS DE POTÊNCIA**

Trabalho Final de Curso apresentado ao Departamento de Engenharia Elétrica da Universidade Federal de Mato Grosso, como requisito parcial para a obtenção do título de Bacharel em Engenharia Elétrica.

Orientador:
Prof. Dr. Carlos Enrique Portugal Poma

CUIABÁ – MT
DEZEMBRO, 2022

Dados Internacionais de Catalogação na Fonte.

F128a Fabris, João Vitor.

Aplicação de redes neurais artificiais à análise de estabilidade de tensão no monitoramento de sistemas elétricos de potência [recurso eletrônico] / João Vitor Fabris. -- Dados eletrônicos (1 arquivo : 73 f., il. color., pdf). -- 2022.

Orientador: Carlos Enrique Portugal Poma.

TCC (graduação em Engenharia Elétrica) - Universidade Federal de Mato Grosso, Faculdade de Arquitetura, Engenharia e Tecnologia, Cuiabá, 2022.

Modo de acesso: World Wide Web: <https://bdm.ufmt.br>.

Inclui bibliografia.

1. Estabilidade de Tensão, Sistema de Energia Elétrica, Redes Neurais Artificiais. I. Poma, Carlos Enrique Portugal, *orientador*. II. Título.

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Permitida a reprodução parcial ou total, desde que citada a fonte.

UNIVERSIDADE FEDERAL DE MATO GROSSO

DESPACHO

Processo nº 23108.103809/2022-11

Interessado: RAUL VITOR ARANTES MONTEIRO, FILLIPE MATOS DE VASCONCELOS, CARLOS ENRIQUE PORTUGAL POMA

FOLHA DE APROVAÇÃO

TÍTULO DA MONOGRAFIA: APLICAÇÃO DE REDES NEURAIAS ARTIFICIAIS À ANÁLISE DE SEGURANÇA DE TENSÃO NO MONITORAMENTO DE SISTEMAS ELÉTRICOS DE POTÊNCIA.

ALUNO: **JOÃO VITOR FABRIS**

Trabalho de Conclusão de Curso apresentado à Faculdade de Arquitetura, Engenharia e Tecnologia da Universidade Federal de Mato Grosso, como requisito para a obtenção de grau de bacharel em Engenharia Elétrica.

Aprovada em 13 de dezembro de 2022.

Nota final: Nove e meio (9,5)

ASSINATURAS:

Banca:

Prof. Dr. Fillipe Matos de Vasconcelos

Prof. Dr. Raul Vitor Arantes Monteiro

Prof. Dr. Carlos Enrique Portugal Poma



Documento assinado eletronicamente por **RAUL VITOR ARANTES MONTEIRO, Docente da Universidade Federal de Mato Grosso**, em 13/12/2022, às 17:07, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **FILLIPE MATOS DE VASCONCELOS**, **Docente da Universidade Federal de Mato Grosso**, em 13/12/2022, às 17:10, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



Documento assinado eletronicamente por **CARLOS ENRIQUE PORTUGAL POMA**, **Docente da Universidade Federal de Mato Grosso**, em 14/12/2022, às 00:28, conforme horário oficial de Brasília, com fundamento no § 3º do art. 4º do [Decreto nº 10.543, de 13 de novembro de 2020](#).



A autenticidade deste documento pode ser conferida no site http://sei.ufmt.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_externo=0, informando o código verificador **5372485** e o código CRC **AF375FAE**.

Eu, **João Vitor Fabris**, autorizo o Departamento de Engenharia Elétrica, assim como a Universidade Federal do Mato Grosso a utilizar e reproduzir meu trabalho, apenas para fins de estudo e pesquisa, sendo expressamente vedado qualquer tipo de reprodução para fins comerciais sem prévia autorização específica.

João Vitor Fabris

João Vitor Fabris

AGRADECIMENTOS

À Universidade Federal de Mato Grosso que, pública e gratuita, me ofereceu a oportunidade de realizar o curso de Engenharia Elétrica.

Ao Professor Carlos Enrique Portugal Poma pelas contribuições, disponibilidade e auxílio nos momentos de dúvidas. E a todos os professores do curso de Engenharia Elétrica.

Aos meus pais Nelci e Ladair, e irmãos Fernando e Naielli, pelo incentivo e apoio incondicional.

À minha companheira Waldelaine R. Hoffmann por todo o carinho e apoio durante os últimos anos.

Aos meus amigos que contribuíram direta e indiretamente.

À banca avaliadora deste trabalho.

RESUMO

FABRIS, J.V. **Aplicação de redes neurais artificiais à análise de segurança de tensão no monitoramento de sistemas elétricos de potência.** 2022. 19f. Trabalho Final de Curso (Graduação em Engenharia Elétrica) Universidade Federal de Mato Grosso. Cuiabá, 2022.

Nos últimos anos, o grande aumento da demanda de potência ativa, associado às limitações do sistema elétrico, o faz operar em condições próximas ao limite operacional, o que pode resultar na incapacidade em lidar com perturbações de tensão, tornando-o instável ao ponto de provocar *Blackouts* parciais ou totais. Neste contexto, diversos métodos foram desenvolvidos para avaliação do nível de segurança de tensão em um sistema elétrico de maneira a prevenir a operação instável. No entanto, estes métodos tradicionalmente requerem um alto esforço computacional. Diante disso, o presente trabalho tem por objetivo colaborar com o desenvolvimento de uma Rede Neural Artificial (RNA) para auxílio na operação de um sistema elétrico, indicando ao operador, índices de estabilidade de tensão de todos os barramentos de carga do sistema, com a grande vantagem de utilizar baixo esforço computacional. Para que a RNA tenha boa capacidade previsão, dadas as entradas de parâmetros do sistema, foi proposto neste trabalho também, uma metodologia para criação de pontos operacionais com aplicação de contingências e cálculo de seus respectivos índices de estabilidade de tensão em cada barra de carga. O índice de estabilidade de tensão utilizado neste trabalho se baseia na matriz jacobiana resultante do método de Fluxo de Potência por Newton-Raphson, o determinante da matriz D' . Os resultados obtidos mostraram a eficiência do modelo proposto em estimar índices tendo como entrada somente o módulo da tensão e ângulo das barras, apresentando uma rápida execução. Os resultados mostram que a RNA está apta a lidar com previsão de índices de estabilidade de tensão com o auxílio de dispositivos de medição, possibilitando ações preventivas ao operador.

Palavras-chave: Estabilidade de Tensão. Sistema de Energia Elétrica. Redes Neurais Artificiais.

ABSTRACT

In recent years, the large increase in demand for active power, associated with restrictions on the electrical system, makes it operate in conditions close to the operational limit, which may result in the inability to deal with voltage disturbances, making it instructive to the point of causing partial or total Blackouts. In this context, several methods have been combined to assess the voltage safety level in an electrical system in order to prevent trouble-free operation. However, these methods traditionally require a high computational effort. In view of this, the present work aims to collaborate with the development of an Artificial Neural Network (ANN) to assist in the operation of an electrical system, indicating to the operator, voltage stability indexes of all the system's load bus, with the great advantage of using the low computational effort. In order for the ANN to have good forecasting capacity, data such as inputs of system parameters, a methodology was also proposed in this work for the creation of operational points with the application of contingencies and calculation of their respective voltage stability indices in each load bus. The voltage stability index used in this work is based on the Jacobian matrix resulting from the Newton-Raphson Power Flow method, the determinant of the D' matrix. The results appreciated the efficiency of the standard model in estimating indices having as input only the voltage module and angle of the bars, presenting a fast execution. The results show that the ANN is able to deal with predictions of voltage stability indices with the aid of measuring devices, providing preventive actions to the operator.

Keywords: Voltage Stability. Electric Power System. Artificial Neural Networks.

LISTA DE ILUSTRAÇÕES

Figura 1 - Os componentes do sistema <i>ADS online</i>	24
Figura 2 - Sistema elétrico de duas barras.	25
Figura 3 - Soluções de tensão na carga com um mesmo fator de potência	26
Figura 4 - Curva para $\phi = 41,19^\circ$ no plano SV.....	27
Figura 5 - Obtenção da margem M pelo método da curva SV.....	28
Figura 6 - Localização dos vetores gradientes de P_i e Q_i no plano θV	30
Figura 7 – Representação do Perceptron de Rosenblatt.	32
Figura 8- Funções de Ativação.....	33
Figura 9 - Rede densa com uma camada oculta e uma camada de saída.	33
Figura 10 - Diagrama de blocos do algoritmo de aprendizado supervisionado.	34
Figura 11 – Exemplo de sistemas com <i>Overfitting</i> , <i>Underfitting</i> , e treinamento ótimo.	36
Figura 12 - Ilustração do método de Validação Cruzada K-fold. Em azul o subgrupo de dados para validação, e em branco, para treinamento.	36
Figura 13 - Curva de validação cruzada genérica em relação ao erro durante o treinamento. .	37
Figura 14 - Sistema de testes IEEE 14-bus.	38
Figura 15- Fluxograma para gerar pontos de operação do sistema IEEE 14-bus.	40
Figura 16a - Gráfico boxplot dos módulos das tensões barras de carga; Figura 17b - Gráfico boxplot dos ângulos das tensões das barras de carga.	45
Figura 18 – Características estatísticas do banco de dados de parâmetros V e θ não normalizados.....	46
Figura 19 - Gráfico em Boxplot do banco de dados dos parâmetros do sistema após normalização.....	47
Figura 20 - Gráfico em Boxplot do banco de dados dos índices de estabilidade de tensão.	47
Figura 21 - Características estatísticas do banco de dados dos índices de estabilidade de tensão $\det[D']$ não normalizados.	48
Figura 22 - Gráfico em Boxplot do banco de dados normalizados dos índices de estabilidade de tensão.....	48
Figura 23 - Erro absoluto médio de treinamento e validação em função das épocas de treinamento.	49
Figura 24 - Representação unifilar do Sistema elétrico de potência IEEE 9-bus.....	56

LISTA DE TABELAS

Tabela 1 - Erro Absoluto médio para treinamento e teste do modelo final.....	50
Tabela 2 - Erro Absoluto médio para treinamento e teste do em validação cruzada K-fold....	50
Tabela 3 - Tipos de barras e incógnitas para a barra “i”.....	57

LISTA DE ABREVIATURAS E SIGLAS

ADS	Avaliação de Segurança Dinâmica
CIGRE	Comitê Internacional de Produção e Transmissão de Energia Elétrica
Det[D']	Determinante da Matriz D'
FVSI	<i>Fast Voltage Stability Index</i>
IEEE	Instituto de Engenheiros Eletricistas e Eletrônicos
Lmn	Índice Lmn
ML	<i>Machine Learning</i>
NLSI	<i>Novel Line Stability Index</i>
RNA	Rede Neural Artificial
RNAs	Redes Neurais Artificiais
PMU	<i>Phasor Measurement Unit</i>
SCADA	<i>System Control and Data Acquisition</i>
SEP	Sistema Elétrico de Potência
SEPs	Sistemas Elétricos de Potência
STATCOM	<i>Static Synchronous Compensator</i>

LISTA DE SÍMBOLOS

λ	Letra grega Lambda minúscula
ϕ	Letra grega Fi minúscula
α	Letra grega Alpha minúscula
θ	Letra grega Theta minúscula
β	Letra grega Beta minúscula
Δ	Letra grega Delta maiúscula
∇	Operador Nabla
\times	Operador Produto Vetorial
\vec{P}	Notação Vetorial

SUMÁRIO

1	INTRODUÇÃO	14
1.1	Problemática.....	14
1.2	Justificativa.....	15
1.3	Objetivos	15
1.3.1	Objetivo geral.....	15
1.3.2	Objetivos específicos	15
1.4	Estado da arte.....	16
2	REFERENCIAL TEÓRICO	23
2.1	Avaliação de segurança e operação <i>online</i> em um SEP	23
2.2	Estabilidade de tensão	24
2.2.1	Caracterização do Fenômeno da Estabilidade de Tensão.....	25
2.2.2	Índice de Avaliação das Condições de Segurança de Tensão $\det[D']$	27
2.3	Redes Neurais Artificiais.....	31
2.3.1	Princípios de Funcionamento e Arquitetura	31
2.3.2	Algoritmo de Aprendizagem.....	34
2.3.3	Avaliação do modelo de RNA	35
3	ESTUDO DE CASO	38
3.1	Descrição do sistema de testes IEEE 14-bus.....	38
3.2	Descrição do <i>hardware</i> e <i>softwares</i> utilizados	39
3.3	Geração do banco de dados para treinamento da RNA.....	39
3.3.1	Obtenção dos Pontos de Operação	40
3.3.1.1	Contingência 1.....	41
3.3.1.2	Contingência 2.....	41
3.3.1.3	Contingência 3.....	42
3.3.2	Cálculo do Índice de Estabilidade de Tensão $\det[D']$.....	42
3.3.3	Pré-processamento do Banco de Dados	42

3.4	Arquitetura da RNA	43
3.5	Treinamento da RNA	43
3.6	Avaliação do modelo de RNA final	44
4	RESULTADOS E DISCUSSÕES	45
4.1	Banco de dados obtido pelo sistema IEEE 14-bus	45
4.2	Treinamento da Rede Neural Artificial	49
4.3	Avaliação do modelo final	49
5	CONSIDERAÇÕES FINAIS.....	51
5.1	Sugestões para trabalhos futuros	51
	REFERÊNCIAS	53
	APÊNDICE A – FLUXO DE POTÊNCIA.....	56
	A.1 Modelagem de Fluxo de Potência.....	56
	A.2 Solução do Problema Pelo Método de Newton Raphson	58
	APÊNDICE B – CÓDIGO FONTE DO SISTEMA DE GERAÇÃO DO BANCO DE DADOS	61
	B.1 Código fonte do arquivo MainFinal.m.....	61
	B.2 Código fonte da função MTP	63
	APÊNDICE C – CÓDIGO FONTE DO ALGORITMO DE PRÉ-PROCESSAMENTO DO BANCO DE DADOS, TREINAMENTO DA REDE NEURAL E VALIDAÇÃO	66
	ANEXO A – DADOS DO SISTEMA IEEE 14-BUS	69

1 INTRODUÇÃO

Historicamente, garantir a operação segura sempre foi de suma importância na operação dos sistemas elétricos de potência. Um sistema elétrico com um grau de segurança insuficiente, torna-se vulnerável a possíveis perturbações, as quais podem chegar a ter consequências catastróficas, tanto em termos de custos econômicos, como também de perda de vidas humanas, como analisado em Veloza e Santamaria (2016).

Para que a operação do sistema elétrico seja segura, este deve permanecer estável. Diz-se que um sistema é estável se para uma dada condição de operação inicial, recupera seu estado de equilíbrio após ser submetido a uma perturbação física, com a maioria das variáveis do sistema dentro de seus limites (KUNDUR, et al., 2004).

Conforme os sistemas elétricos evoluíram em escala e complexidade, o uso de novas técnicas de controle, novas características de carga, e submetidos cada vez mais a condições intensas de operação, houve o aparecimento de diversas formas de instabilidades ao sistema, que são avaliadas em três grandes áreas de estudo: estabilidade do ângulo do rotor, estabilidade de tensão e estabilidade de frequência (KUNDUR, 1994). Recentemente foram adicionadas mais duas grandes áreas de estudo de estabilidade, a estabilidade de ressonância e estabilidade orientada por conversor (HATZIARGYRIOU et al., 2021).

O foco deste trabalho está no desenvolvimento de um algoritmo computacional para geração de um conjunto de pontos operacionais com diferentes tipos de contingência de um sistema elétrico e seus respectivos níveis de estabilidade de tensão, e um modelo de Rede Neural Artificial utilizando técnicas de Regressão. O modelo proposto foi desenvolvido no decorrer deste trabalho e poderá ser aplicado na operação de um sistema elétrico para avaliação de segurança no âmbito da estabilidade de tensão.

1.1 Problemática

A operação bem sucedida de um sistema elétrico de potência depende fortemente do fornecimento de um serviço confiável e sem interrupção do fornecimento de energia elétrica para as cargas (ANDERSON; FOUAD, 2003). Segundo Morison, Wang e Kundur (2004) para garantir que o sistema seja suficientemente confiável, o mesmo deve ser projetado tendo a segurança como principal objetivo, e monitorado durante a operação para garantir que exista margem de segurança suficiente em todos os momentos, em frequência e tensão.

Diante disso, este trabalho busca responder os questionamentos:

- A operação do sistema elétrico é confiável e segura quando próximo do ponto de máximo de carregamento?

- Quais as consequências do aumento ou diminuição progressiva da potência no fornecimento de energia elétrica?
- Como é feito o monitoramento de um sistema elétrico de potência no que se refere a avaliação de segurança?
- Os atuais métodos utilizados para verificação dos níveis de estabilidade de tensão são viáveis na operação de um sistema elétrico de potência?
- Existem alternativas aos tradicionais métodos computacionais de previsão de estabilidade em um sistema elétrico?

1.2 Justificativa

O sistema elétrico pode ser submetido a um processo pelo qual, uma sequência de eventos podem desencadear um apagão, ou apresentar fora dos limites permitidos. Este fenômeno pode ser originado pelo uso de linhas de transmissão perto de sua capacidade máxima. Durante a operação do sistema elétrico não é recomendável utilizar os tradicionais métodos de análise baseados na construção de curvas PV e QV, uma vez que sua obtenção é demorada. Por tanto, o uso de um método de inteligência artificial baseado em aprendizado de máquina para obtenção de índices de estabilidade é uma alternativa mais rápida para que, a partir dos índices de estabilidade sejam selecionadas as barras críticas do sistema de testes IEEE 14-bus.

1.3 Objetivos

1.3.1 Objetivo geral

Aplicar Redes Neurais Artificiais na avaliação da segurança de tensão para cálculo de índices de estabilidade de tensão em um sistema elétrico, visando aplicação em sistemas de monitoramento *online*.

1.3.2 Objetivos específicos

- a) Fundamentar os conceitos que regem a análise de segurança na operação de um sistema elétrico de potência em tempo real, o estudo da estabilidade de tensão, e redes neurais artificiais;
- b) Analisar e comparar diferentes bibliografias sobre tipos de aprendizado de máquina voltado à sistemas de monitoramento *online* em estabilidade de tensão;

- c) Aplicar métodos de fluxo de potência eficientes disponíveis em pacotes de *software* livre em Matlab, para extração dos parâmetros de um sistema elétrico para as mais variadas contingências, além de implementar o cálculo de índices de estabilidade de tensão para cada caso;
- d) Implementar, treinar e validar uma rede neural artificial utilizando ferramentas específicas em linguagem de programação Python;
- e) Avaliar a aplicabilidade da rede neural artificial desenvolvida na operação *online* de um sistema elétrico de potência.

1.4 Estado da arte

Segundo Haykin (2001), uma rede neural artificial é um modelo computacional que procura imitar a estrutura paralela e maciça de pequenas unidades de processamento vistas em um sistema biológico. Este sistema é capaz de adquirir conhecimento experimental e possui alta capacidade de generalização, que se refere a capacidade do sistema de produzir uma saída adequada para entradas não vistas anteriormente. Estas capacidades permitem ao sistema resolver problemas complexos de grande escala, que normalmente são intratáveis (HAYKIN, 2001).

Muitos trabalhos foram desenvolvidos utilizando redes neurais artificiais como alternativa rápida aos tradicionais métodos de solução do problema de estabilidade de tensão em um sistema elétrico de potência, que exigem grande esforço computacional à medida que o sistema aumenta em tamanho e complexidade.

De acordo com Zhou, Annakkage e Rajapakse (2010), uma das principais ferramentas empregadas em análises de estabilidade de tensão em uma barra isolada do sistema elétrico é a análise da curva PV para obtenção da margem de potência. Em um grande sistema de monitoramento e controle *online*, este método pode se tornar inviável devido ao grande esforço computacional. Neste artigo, foi proposto o desenvolvimento de quatro modelos de rede neural artificial do tipo *Feedforward* para previsão da margem de estabilidade de tensão em barras isoladas, onde avalia-se o desempenho de cada modelo com relação aos tipos de dados utilizados em seu treinamento.

Para a geração do banco de dados, o autor utilizou como base os sistemas elétricos *New England 39-bus* e *Alberta 1844-Bus System*. No primeiro, com um incremento de $\pm 30\%$ como caso inicial, prosseguiu-se com fluxo de potência continuado para estimar a margem de potência por curvas PV, obtendo um total de 3000 casos para treinamento do modelo proposto, e além disso, 100 casos foram gerados e separados para teste do modelo final. Para o sistema *Alberta*

1844-Bus foi obtido também a mesma quantidade de pontos operacionais para treinamento e testes, com o diferencial de serem gerados pelo *software* comercial da *Powertech*, o *DSATools*. A margem de potência calculada pelo módulo *VSAT* do *software*, teoricamente difere do calculado no sistema *New England 39-bus*, pois retorna um limite de potência reativa à barra, porém é válida para a análise de estabilidade de tensão.

Os modelos de Redes Neurais do tipo *Feedforward* foram treinados pelo método de aprendizado por retropropagação, onde os dados de entrada utilizados para cada modelo foram:

1. magnitude das tensões;
2. Injeções líquidas de potência ativa e reativa;
3. magnitude de tensão e potência reativa e
4. magnitude de tensão e ângulo de fase.

A performance de cada modelo foi avaliada, onde na quarta opção o erro foi o menor obtido. Com isso, conclui-se que o modelo pode ser utilizado para aplicação em monitoramento *online* em conjunto com obtenção de dados do sistema elétrico por PMUs sincronizados, que a cada instante de tempo aferem a medição de tensão e ângulo sob um ponto de referência.

Em Zhang et al., (2013) é proposto a aplicação de um novo paradigma de aprendizado de máquina em Redes Neurais, o *Extreme Learning Machine* (ELM). No entanto, aprendizado por ELM pode não ter uma acurácia tão boa, então, neste trabalho desenvolve-se um agrupamento de ELMs, com o objetivo de atenuar os erros de predição do modelo final. Além disso, pela preocupação com a inserção crescente do *Smart Grid* e participação da geração do lado da demanda, é incluído para o treinamento dos modelos propostos, a direção do carregamento para obtenção do banco de dados.

Para fins de treinamento-teste dos modelos propostos, o sistema IEEE 118-bus foi utilizado para geração do banco de dados, onde os pontos operacionais iniciais foram obtidos assumindo variações de carga de 80% a 120% de seu valor nominal, com base nas equações: $P_{Li} = P_{Li0} + \lambda \times (d_i \times P_{Li0})$ e $Q_{Li} = Q_{Li0} + \lambda \times (d_i \times Q_{Li0})$, onde d_i é a direção da carga na i -ésima barra de carga do sistema. A partir destes dados iniciais, foi produzido pontos de operação do sistema com fluxo de potência para posterior cálculo da margem de estabilidade por CPF. Esta etapa gerou uma grande quantidade de dados, somando 22900 pontos operacionais, onde 80% foram destinados a treinamento e 20% para teste do modelo ELM.

Os dados foram utilizados para treinamento-teste em diferentes tipos de treinamento, para comparação: uma única ELM, um agrupamento de 100 ELMs e uma rede treinada pelo tradicional método de retropropagação. O desempenho de cada modelo, em sequência foram, 4.77%, 3.32% e 4.63% e o tempo de treinamento foi de 0.074s, 7.16s e 336.2s.

Em Suganyadevi e Babulal, (2014), uma nova metodologia é desenvolvida com base em aprendizado por máquinas de *Support Vector Regression* (SVR), uma aplicação mais comum de *Support Vector Machines* (SVM). As SVRs procuram estimar uma função que mapeia a partir de um objeto de entrada, um número real de saída.

A eficiência de quatro modelos de SVR's são avaliados, sendo dois do tipo ν -SVR, um com *kernel* com função de base Radial, e um Kernel com função polinomial, e os mesmos tipos para a ϵ -SVR. Ao fim, compara-se todos os modelos propostos com uma rede neural treinada por retropropagação.

Os sistemas IEEE 30-bus e IEEE 118-bus foram utilizados para geração de pontos operacionais com fluxo de potência continuado pelas ferramentas PSAT e VSAT, da *DSATools*TM. Os dados utilizados para treinamento e teste dos modelos foram as potências ativas e reativas líquidas a barra, além da margem de carregamento. No total foi gerado 36000 dados para o sistema IEEE 30-bus, e 56640 para o sistema IEEE 118-bus, onde 80% foi destinado para treinamento, e 20% para validação. Ao fim, os autores utilizaram métodos de validação cruzada para verificar o poder de generalização dos modelos.

Os modelos de SVR apresentaram melhores resultados em comparação a ANN, tanto em tempo computacional quanto em precisão na predição. O modelo ϵ -SVR obteve uma melhor performance em relação ao ν -SVR, com baixíssimo erro e tempo computacional, sendo capaz de trabalhar em um sistema *online* de controle.

Com o recente aumento da integração de parques eólicos com o sistema elétrico, é proposto em Duraipandy et al., (2017), uma Rede Neural treinada com *Extreme Machine Learning* (ELM) para previsão de margem de carregamento do sistema considerando entradas de parâmetros gerados pelos dispositivos de conversão das turbinas eólicas. O autor utiliza a análise dinâmica neste trabalho, avaliando a estabilidade em tensão transitória na barra em análise.

O regulador de tensão automático e a turbina são modelados por meio de equações diferenciais considerando variáveis como a velocidade do vento, densidade do ar, área varrida pelas pás, etc. O sistema elétrico utilizado foi o IEEE 30-bus, com a inserção de turbina eólica na barra 30, onde foi encontrado uma maior margem de estabilidade com fluxo de potência continuado estático.

O banco de dados é gerado a partir do ponto inicial com carregamento num alcance de 75% a 125% do valor nominal, enquanto a velocidade do vento também varia de 14m/s a 25m/s para entrada no DCPF, que calculará as margens de estabilidade de tensão para cada caso em

específico. 500 diferentes condições de carregamento foram feitas, onde 375 foram destinadas a treinamento, e 125 a testes.

A precisão dos resultados do modelo proposto de ELM foi bastante alta, evidenciando sua superioridade a uma rede BPNN usada para comparação, provando ser uma excelente alternativa para monitoramento *online*.

Índices de estabilidade de tensão indicam a distância do ponto de operação atual do sistema até o ponto de colapso de tensão, e em Usama et al., (2017), é proposto o uso de tais índices para treinamento e validação de dois tipos de aprendizado de máquina, uma Rede Neural (RNA) e uma Árvore de Decisão, e também é analisado a técnica estatística de Regressão Linear, para o problema de estabilidade de tensão. A melhor aplicação fará parte de um sistema de alerta chamado *Voltage Stability Automatic Maneuver Algorithm* (V.S.A.M.A), que em tradução literal, chama-se Algoritmo de Manobra Automática de Estabilidade de Tensão. Este Sistema avisará o operador sobre o status da operação, se está ou não com níveis de segurança aceitáveis, e também, entrega ao operador, possíveis soluções para o problema, avaliando cada possibilidade possível.

O banco de dados para o treinamento do algoritmo foi obtido por meio de simulações computacionais com o *software* Power World Simulation Tool, utilizando como base o sistema IEEE 9-bus, gerando 1300 diferentes exemplos contendo variações de potências ativas e reativas nas cargas e geradores. Tais dados foram utilizados para cálculo de três diferentes índices de estabilidade de tensão: FVSI, Lmn e NLSI.

Três bancos de dados foram gerados para treinamento-teste dos três modelos propostos, ambos com os mesmos dados, exceto os valores dos índices. Os autores avaliaram a precisão dos modelos e a melhor combinação obtida foi a previsão utilizando Redes Neurais para o índice NLSI, obtendo um erro de Raiz Quadrada Média de 0,08.

O sistema V.S.A.M.A foi testado escolhendo um instantâneo do Sistema IEEE 9-bus em que o sistema alertou que a barra 4 estava operando de maneira insegura. A partir disso, o V.S.A.M.A sugeriu dez diferentes condições de carregamento avaliando com a RNA, a estabilidade geral do sistema elétrico. O melhor cenário de operação foi evidenciado como sendo o quinto, onde todas as linhas estariam estáveis e o problema de instabilidade de tensão seria resolvido.

Em Wokoma, Idachaba e Osegi, (2019), métodos de treinamento de máquina baseados em RNA, Redes Neurais Recorrentes (RNN), Sistemas de Múltiplos Agentes foram propostos para prever valores de índices de estabilidade de tensão até a elaboração deste trabalho, e grande

parte sofre de desarmonia estocástica na previsão da margem de estabilidade do sistema, além de necessitarem de atualização constante de hiper-parâmetros.

Neste trabalho, é utilizado uma técnica inteligente de nome, em inglês, Auditory Machine Intelligence (AMI) para previsão de índices de estabilidade de tensão em linhas de transmissão q-LVSI (do inglês, *Quadratic Line Voltage Stability Index*). AMI consiste em utilizar processos determinísticos dependentes do tempo para previsão de séries temporais (OSEGI; ANIREH, 2019) e tem algumas vantagens em relação aos métodos propostos em trabalhos anteriores para previsão de estabilidade de tensão. O método foi comparado com o método Método de Tratamento de Dados Agrupados (abreviado no inglês de GMDH), que é o estado-da-arte a respeito de previsões usando algoritmos de ML com séries temporais.

Todos os experimentos foram realizados em Matlab, com a modelagem do sistema elétrico de potência nigeriano de três barras em 330kV, e fluxo de potência por método iterativo de Newton Raphson, gerando 10 pontos operacionais.

Os resultados obtidos para os dois modelos de aprendizado de máquina foram avaliados utilizando o Erro Absoluto Médio Percentual. O método AMI se apresentou superior em somente uma das três barras, porém nas demais ocasiões, a diferença é mínima, mostrando ser uma técnica competitiva ao método GMDH para previsão de índices de estabilidade de tensão.

Em Adhikari et al., (2020) compara-se quatro métodos de *Machine Learning* baseados em regressão para prever a margem de estabilidade de tensão: um Método de Processo Gaussiano (MPG), Máquina de Suporte ao Vetor (SVM), Rede Neural Artificial (RNA) e Árvore de Decisão (AD), diferenciando-se de outros autores por incluir novos tipos de contingências ao sistema.

Muitos estudos comparam algoritmos de RNA, SVM e AD com base em poucos tipos de contingências (somente aumento de carga em componentes do sistema em um determinado alcance), o que para aplicações *online* e reais não seriam uma boa opção. O autor introduz condições desconhecidas no teste dos modelos, como a remoção de uma linha de transmissão ou gerador, e compara com simples aumentos de carga.

No trabalho, a base de dados foi gerada por meio de fluxo de potência continuado utilizando como base o sistema elétrico New England 39-bus, utilizando para treinamento dos modelos, apenas dados de tensão (em P.U.) e ângulo da tensão, sendo essa escolha motivada pelo uso de PMUs em sistemas de monitoramento e controle. O índice de estabilidade de tensão adotado foi a margem M, obtida por fluxo de potência continuado. Para a contingência de aumento de carga, foi feita uma variação de carga do sistema entre 0,8 e 1,2, gerando 4461

pontos de operação, enquanto para a contingência de eliminação de linha de transmissão e gerador, foi obtido 2337 pontos operacionais.

O banco de dados foi dividido em 70% para treinamento, e 30% para validação/teste de ambos os modelos. Para a primeira condição de treinamento-teste, utilizou-se os 4461 casos de aumento de carga somente, e para a segunda condição, o total de dados, abrangendo as duas contingências. A comparação de erros obtidos mostrou que o modelo MPG apresentou melhor desempenho para a primeira condição, enquanto os demais modelos, para a segunda condição.

Os modelos MPG e ANN necessitaram de um grande tempo de treinamento em comparação com SVM e AD, embora MPG e RNA, respectivamente, apresentaram uma melhor performance, cujos resultados devem ser avaliados para possível aplicação, seja em monitoramento *online*, ou estudos *off-line*.

Em Dharmapala et al. (2020), os autores propõem o uso de diferentes algoritmos de ML baseados em regressão para prever a margem de carregamento do sistema elétrico de longo prazo, que serve como um indicador para estabilidade de tensão, cujo diferencial de outras aplicações vistas até o momento, é o uso de um conjunto de índices de estabilidade de tensão, cada um com suas características, e sincrofasores como dados de entrada para treinamento dos algoritmos de *Machine Learning*, e a margem de carregamento como saída.

Os tipos de modelos de inteligência artificial utilizados foram: Regressão Linear (RL); Regressão Polinomial (PR); Gradient Boost Regression (GBR); Rede neural Artificial (RNA) e Random Forest Regression (RFR). Quatro agrupamentos distintos de diferentes índices de estabilidade de tensão, de acordo com suas características, foram utilizados para treinamento-teste de modo a verificar a melhor opção.

A base de dados foi gerada com base nos sistemas IEEE 14-bus e IEEE 118-bus por meio do *software* PSSE®, o que resultou em 3.196 e 5.424 dados respectivamente por meio de uma seleção de dados obtidos por fluxo de potência continuado, que posteriormente serviu para cálculo dos índices de estabilidade de tensão.

O melhor resultado obtido foi com a rede *Random Forest Regression* (RFR), que se adaptou bem a todos os agrupamentos de dados de treinamento-teste. Observou-se também que para sistemas maiores, o agrupamento com maior número de índices de estabilidade de tensão performou melhor ao algoritmo de aprendizado de máquina.

Em Queen et al., (2021) é desenvolvido dois modelos de ML para predição técnico-econômica em termos da estabilidade de tensão na tomada de decisões do operador. Além disso, é feita a inserção de energias renováveis e dispositivos STATCOM a dois sistemas de teste, visando contribuir para uma operação rápida e confiável em um sistema de energia moderno.

Os modelos de aprendizado de máquina utilizados foram de Regressão Linear e polinomial de ordens 3, 4 e 5.

A geração do conjunto de pontos operacionais se deu com o acréscimo de fontes de energia renováveis de uma planta solar de 25MW e uma planta eólica de 50MW, além de um dispositivo STATCOM com base na inserção em um dia (24hrs). O fluxo de potência ótimo, que considera o custo da operação foi realizado, com método iterativo de Newton-Raphson em Matlab sobre os sistemas teste IEEE 14-bus e IEEE 30-bus. O índice de estabilidade de tensão adotado é o índice L, que relaciona a matriz de admitâncias com a tensão do barramento.

As previsões de índice de estabilidade com considerando regressão linear e polinomial de diferentes ordens foram avaliadas, bem como validação cruzada do tipo *K-fold* considerando $K=5$, onde obteve-se um erro para todos os modelos, superior a 0,9 em R^2 , tanto para a previsão do índice L de estabilidade de tensão, quanto para os valores econômicos.

2 REFERENCIAL TEÓRICO

Esta seção busca fundamentar sobre a operação segura de um sistema elétrico, bem como a teoria da existência de um máximo carregamento à uma barra de carga. Além de maneiras de se quantificar os níveis de estabilidade de tensão por meio de índices. Além disso, os conceitos que regem o funcionamento de Redes Neurais Artificiais (RNA), os tipos de arquiteturas e suas formas de treinamento e validação são abordadas a frente.

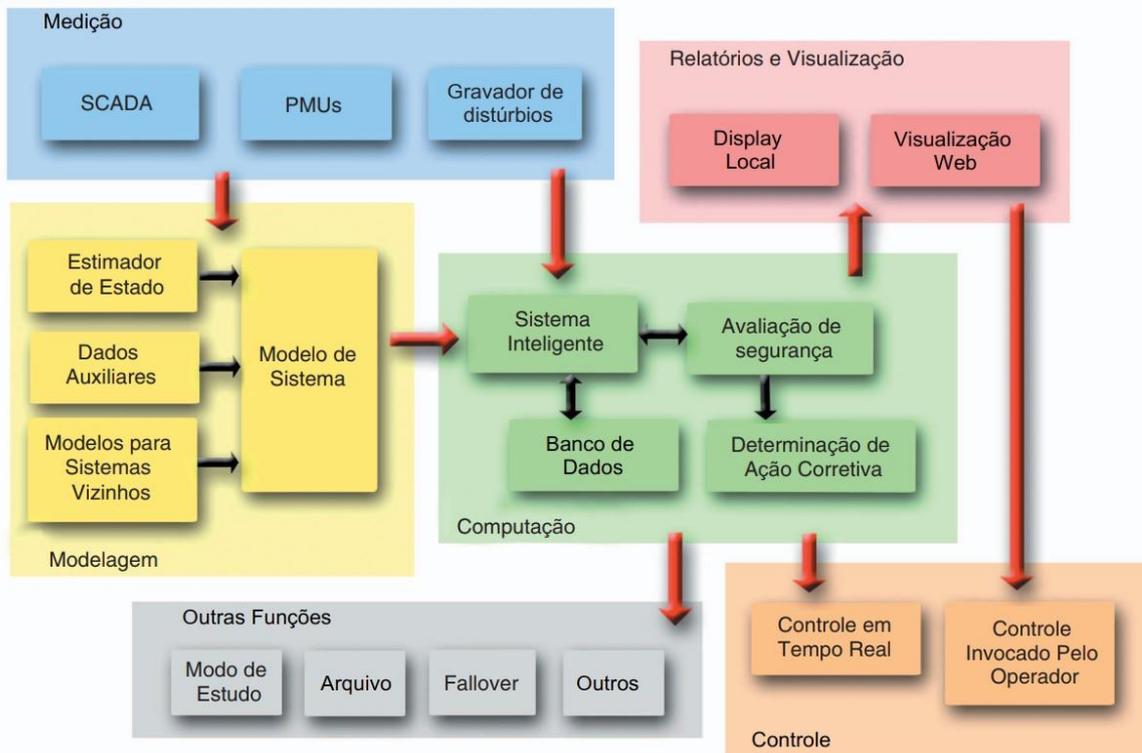
2.1 Avaliação de segurança e operação *online* em um SEP

Segurança refere-se ao grau de risco na capacidade de um sistema elétrico sobreviver a perturbações iminentes sem interrupção do atendimento ao cliente, que depende diretamente da condição de operação do sistema e da probabilidade contingente de distúrbios (MORISON; WANG; KUNDUR, 2004).

Morison et al. (2004) define o termo ADS (Avaliação de Segurança Dinâmica), que se refere à análise necessária para determinar se um sistema de energia pode ou não atender os critérios de confiabilidade e segurança em condições de pré-contingência e pós-contingência para abordagens em regime permanente e regime transitório. Isso implica em avaliá-lo em todos os aspectos de segurança, tais como:

- Carga térmica dos elementos de transmissão;
- Flutuações de tensão e frequência em regime permanente;
- Queda ou aumento progressivo de tensão;
- Estabilidade transitória;
- Estabilidade de tensão;
- Estabilidade de frequência.

Como alternativa ao antigo sistema de operação *offline* com exaustivos fluxos de potência e simulações no domínio do tempo, que se tornou insuficiente dada a complexidade dos atuais sistemas elétricos, Morison et al. (2004) propôs ainda um sistema de monitoramento *online*, o *ADS online*, que é mostrado na Figura 1.

Figura 1 - Os componentes do sistema *ADS online*.

Fonte: Traduzido de Morison et al., (2004).

No sistema de medição, têm-se sistemas de aquisição de dados SCADA (*System Control and Data Acquisition*), dispositivos sincronizados para medição de módulo da tensão e fase PMU's (*Phasor Measurement Unit*) que passam por um pré-processamento, e gravador de distúrbios. A modelagem do sistema é feita para avaliação das condições de segurança, que são repassadas ao sistema computacional que irá identificar e alertar qualquer anormalidade e possível solução para os operadores do sistema via *Web* ou *Display Local*. Os conceitos que regem a determinação dos níveis de segurança de um sistema elétrico são abordados a seguir.

2.2 Estabilidade de tensão

A capacidade do sistema elétrico de manter tensões constantes em todas as barras do sistema sob condições normais de operação e após ser submetida a uma perturbação define um sistema estável em tensão (KUNDUR, 1994).

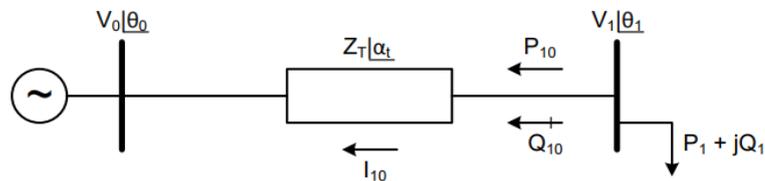
A instabilidade pode ocorrer na forma de uma queda ou aumento progressivo da tensão de alguns barramentos, que segundo Kundur (1994), são resultados de perdas de potência reativa e quedas de tensão que ocorrem quando altos níveis de potência ativa e reativa fluem através de reatâncias indutivas associadas à rede de transmissão, limitando sua capacidade de transferência de energia. Portanto, a existência de uma máxima carga que pode ser alimentada

pela rede de transmissão é a manifestação mais conhecida do problema de estabilidade de tensão.

2.2.1 Caracterização do Fenômeno da Estabilidade de Tensão

Para compreensão do fenômeno de estabilidade de tensão, será abordado o comportamento estático de um sistema elétrico de duas barras (CID, 2013), devido a simplicidade e capacidade de generalização para um sistema de múltiplas barras. Considera-se para este exemplo um gerador com capacidade infinita de geração, uma carga modelada por potência constante e uma linha de transmissão sem limite térmico. Os valores das admitâncias *shunt* da linha de transmissão serão desprezados sem perda de generalização.

Figura 2 - Sistema elétrico de duas barras.



Fonte:(CID, 2013).

A caracterização do problema é desenvolvida considerando:

$$\begin{aligned}
 V_0 &= 1 \text{ pu} \\
 \theta_0 &= 0^\circ \\
 Z_t &= 0.2 \text{ pu} \\
 \alpha_t &= 70^\circ \\
 -\infty &< P_0 < +\infty \\
 -\infty &< Q_0 < +\infty
 \end{aligned} \tag{2.1}$$

O fluxo de potência aparente que sai da barra de carga é dado por:

$$S_{10}^* = P_{10} - jQ_{10} = V_1^* I_{10} \tag{2.2}$$

Onde

$$I_{10} = \frac{V_1 \angle \theta_1 - V_0 \angle \theta_0}{Z_t \angle \alpha_t} \tag{2.3}$$

$$V_1^* = V_1 \angle -\theta_1 \tag{2.4}$$

Substituindo 2.3 e 2.4 em 2.2 obtém-se:

$$S_{10}^* = \frac{V_1^2 \cos(\alpha_t)}{Z_t} - \frac{V_1 V_0 \cos(\theta_{10} + \alpha_t)}{Z_t} - j \left[\frac{V_1^2 \sin(\alpha_t)}{Z_t} - \frac{V_1 V_0 \sin(\theta_{10} + \alpha_t)}{Z_t} \right] \tag{2.5}$$

Comparando 2.5 e 2.2, pode-se separar as partes real e imaginária:

$$P_{10} = -P_1 = \frac{V_1^2 \cos(\alpha_t)}{Z_t} - \frac{V_1 V_0 \cos(\theta_{10} + \alpha_t)}{Z_t} \quad (2.6)$$

$$Q_{10} = -Q_1 = \frac{V_1^2 \sin(\alpha_t)}{Z_t} - \frac{V_1 V_0 \sin(\theta_{10} + \alpha_t)}{Z_t} \quad (2.7)$$

A tangente do ângulo do fator de potência na carga para esse sistema é:

$$\tan(\phi) = \frac{Q_{10}}{P_{10}} = \frac{\frac{V_1^2 \sin(\alpha_t)}{Z_t} - \frac{V_1 V_0 \sin(\theta_{10} + \alpha_t)}{Z_t}}{\frac{V_1^2 \cos(\alpha_t)}{Z_t} - \frac{V_1 V_0 \cos(\theta_{10} + \alpha_t)}{Z_t}} \quad (2.8)$$

Rescrevendo as equações 2.6, 2.7 e 2.8:

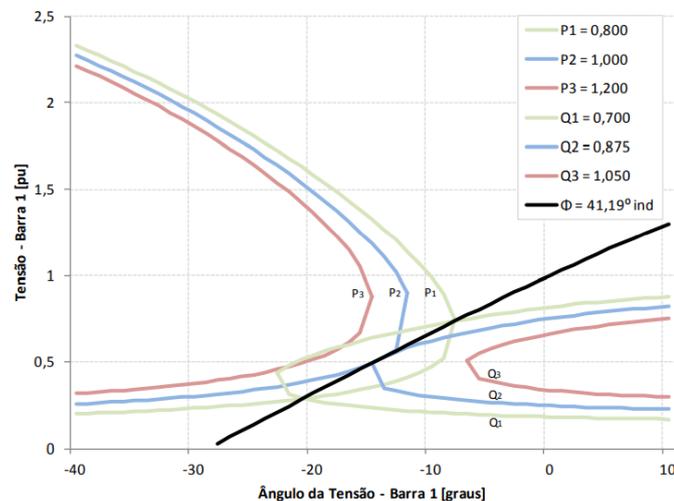
$$V_1^2 \left[\frac{\cos(\alpha_t)}{Z_t} \right] - V_1 \left[\frac{V_0 \cos(\theta_{10} + \alpha_t)}{Z_t} \right] + [P_1] = 0 \quad (2.9)$$

$$V_1^2 \left[\frac{\sin(\alpha_t)}{Z_t} \right] - V_1 \left[\frac{V_0 \sin(\theta_{10} + \alpha_t)}{Z_t} \right] + [Q_1] = 0 \quad (2.10)$$

$$V_1 = V_0 \frac{\sin(\theta_{10} + \alpha_t) - \tan(\phi) \cos(\theta_{10} + \alpha_t)}{\sin(\alpha_t) - \tan(\phi) \cos(\alpha_t)} \quad (2.11)$$

Para diferentes valores de P e Q constantes e um fator de potência de $41,19^\circ$ indutivo, nas equações 2.9 e 2.10, são traçadas num plano $V\theta$, conforme a Figura 3. A curva ϕ constante no plano $V\theta$ é traçada variando-se θ_1 em 2.11 e calculando V_1 .

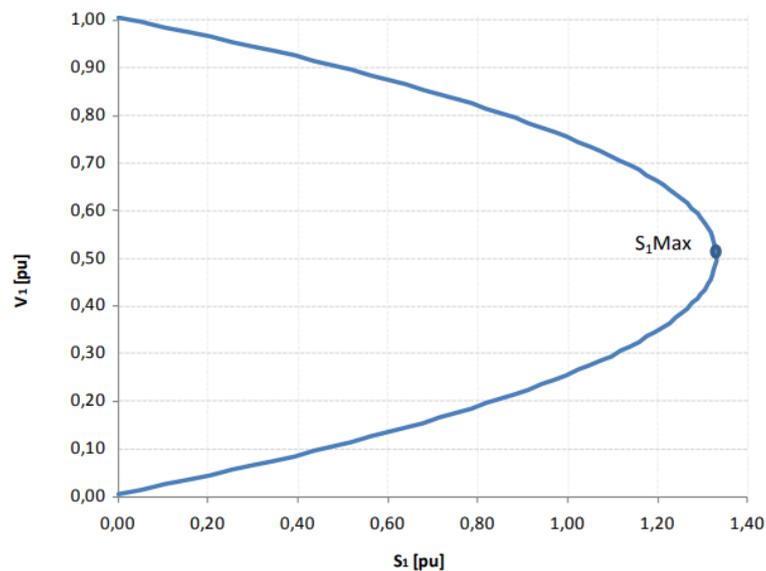
Figura 3 - Soluções de tensão na carga com um mesmo fator de potência



Fonte: Cid, (2013).

Por meio da Figura 3, percebe-se que para um dado fator de potência pode haver mais de uma solução para V_1 e θ_1 . No cenário de carregamento $P_1 - Q_1$ têm-se duas soluções possíveis (duas interseções entre as curvas $P_1 - Q_1$ e ϕ). Aumentando a carga mantendo-se o mesmo fator de potência, para o carregamento $P_2 - Q_2$ a solução é única para o problema, enquanto no cenário $P_3 - Q_3$ não há soluções reais de módulo da tensão e ângulo, mostrando um cenário de carregamento impossível para a barra em análise. Ainda, traçando a curva V_1 em função do carregamento S_1 para o fator de potência supracitado (Figura 4), têm-se o que é conhecido como curva SV, onde o ponto de máximo carregamento é conhecido como “nariz” da curva SV.

Figura 4 - Curva para $\phi = 41,19^\circ$ no plano SV.



Fonte: Cid, (2013).

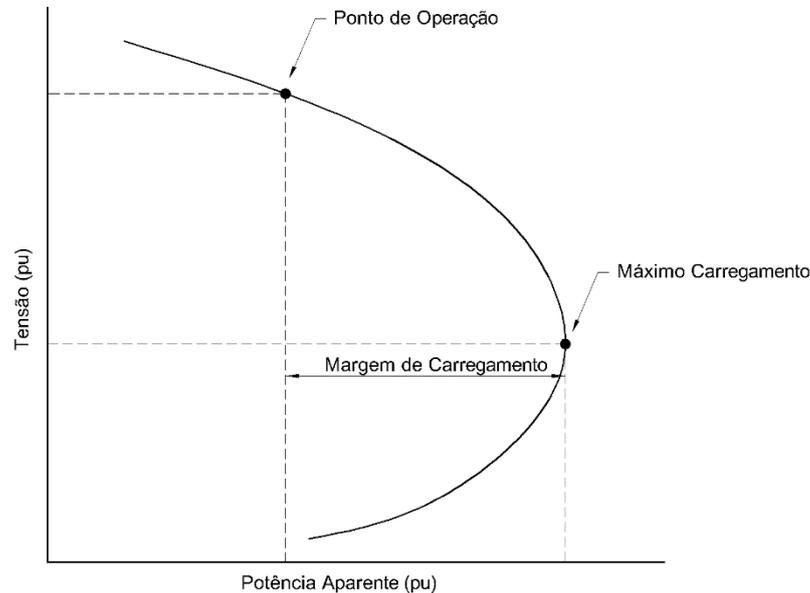
Na área da estabilidade de tensão, diz-se que todos os pontos de operação da barra em análise situados na região acima do ponto de máximo carregamento, e abaixo do mesmo, são pontos de operação estáveis e instáveis, respectivamente (KUNDUR, 1994).

2.2.2 Índice de Avaliação das Condições de Segurança de Tensão $\det[D']$

Diversos métodos matemáticos permitem identificar se a barra em análise está operando na região superior, inferior, ou então a distância do “nariz” da curva SV (ZAHEB et al., 2020). O método mais tradicional e utilizado para tal consiste na análise de curvas SV (ou também PV, QV), conforme visto na Figura 4. Neste método, realizam-se múltiplos fluxos de potência adaptados, que possibilitam uma solução para ambas as regiões estáveis e instáveis (método de fluxo de potência continuado), obtendo-se um máximo ponto de carregamento para uma barra específica, onde a partir daí, é possível definir para um ponto de operação específico, a margem

de potência admissível à barra, conforme a Figura 5. Uma vantagem deste método está no poder de convergência de pontos operacionais na região instável.

Figura 5 - Obtenção da margem M pelo método da curva SV.



Fonte: Adaptado de Dharmapala et al., (2020)

Um método alternativo à margem M é conhecido como magnitude do determinante da matriz $[D']$ (PRADA et al., 2002), cujo valor indica a distância do ponto de máximo carregamento. Este método, bem como outros índices de estabilidade de tensão, foi implementado no programa EstabTen, desenvolvido pela PUC-Rio em convênio com o CEPTEL para o programa de simulação estática ANAREDE.

Para a análise em um sistema contendo duas barras conforme a Figura 2, pode-se através do método iterativo de Newton Raphson, obter as equações estáticas do fluxo de carga para a solução final convergida (ver equação A.12, do APÊNDICE A):

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = [J] \begin{bmatrix} \Delta \theta \\ \Delta V \end{bmatrix} \quad (2.12)$$

Separando as equações e variáveis relacionadas a barra em análise (barra i) na extremidade inferior direita, fica:

$$\begin{bmatrix} \Delta P'_i \\ \Delta Q'_i \\ \Delta P_i \\ \Delta Q_i \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \Delta \theta'_i \\ \Delta V'_i \\ \Delta \theta_i \\ \Delta V_i \end{bmatrix} \quad (2.13)$$

As submatrizes $[A]$, $[B]$, $[C]$ e $[D]$ são partições da matriz Jacobiano $[J]$.

Como o interesse é avaliar variações infinitesimais de carga em ΔP_i e ΔQ_i e seus efeitos sobre ΔV_i e $\Delta \theta_i$ em um sistema multi-nó, supõem-se que $\Delta P' = \Delta Q' = 0$, ou seja, não há variações nas injeções de potência nas outras barras.

$$\begin{bmatrix} 0 \\ 0 \\ \Delta P_i \\ \Delta Q_i \end{bmatrix} = \begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} \Delta \theta' \\ \Delta V' \\ \Delta \theta_i \\ \Delta V_i \end{bmatrix} \quad (2.14)$$

Fazendo todas as operações necessárias, reduz-se a equação 2.14 em:

$$\begin{bmatrix} \Delta P_i \\ \Delta Q_i \end{bmatrix} = [D - CA^{-1}B] \begin{bmatrix} \Delta \theta_i \\ \Delta V_i \end{bmatrix} \quad (2.15)$$

Ou então,

$$\begin{bmatrix} \Delta P_i \\ \Delta Q_i \end{bmatrix} = [D'] \begin{bmatrix} \Delta \theta_i \\ \Delta V_i \end{bmatrix} \quad (2.16)$$

A matriz $[D']$, no entanto, leva em consideração o restante do sistema e possui dimensão 2×2 .

$$[D'] = [D] - [C][A^{-1}][B] \quad (2.17)$$

Ou então, em termos das variações de potência ativa e potência reativa,

$$[D'] = \begin{bmatrix} \frac{\partial P_i}{\partial \theta_i} & \frac{\partial P_i}{\partial |V_i|} \\ \frac{\partial Q_i}{\partial \theta_i} & \frac{\partial Q_i}{\partial |V_i|} \end{bmatrix} \quad (2.18)$$

Aplicando o determinante na matriz $[D']$ têm-se que:

$$\det[D'] = \frac{\partial P_i}{\partial \theta_i} \frac{\partial Q_i}{\partial |V_i|} - \frac{\partial P_i}{\partial |V_i|} \frac{\partial Q_i}{\partial \theta_i} \quad (2.19)$$

Os vetores gradientes de potência ativa e reativa na barra i podem ser expressos em sistema de referência cartesiano:

$$\vec{\nabla P} = \frac{\partial P_i}{\partial \theta_i} \vec{\theta} + \frac{\partial P_i}{\partial |V_i|} \vec{V} + 0 \vec{S} \quad (2.20)$$

$$\vec{\nabla Q} = \frac{\partial Q_i}{\partial \theta_i} \vec{\theta} + \frac{\partial Q_i}{\partial |V_i|} \vec{V} + 0 \vec{S} \quad (2.21)$$

O produto vetorial:

$$\vec{\nabla}P \times \vec{\nabla}Q = \left[\left(\frac{\partial P_i}{\partial \theta_i} \frac{\partial Q_i}{\partial |V_i|} \right) - \left(\frac{\partial P_i}{\partial |V_i|} \frac{\partial Q_i}{\partial \theta_i} \right) \right] \vec{s} \quad (2.22)$$

Ou então,

$$|\vec{\nabla}P \times \vec{\nabla}Q| = |\det[D']| \quad (2.23)$$

Logo,

$$|\nabla P \times \nabla Q| = |\det[D']| = |\dot{\nabla}P| |\dot{\nabla}Q| \text{sen}(\beta) \quad (2.24)$$

Onde β é o ângulo entre $|\nabla P|$ e $|\nabla Q|$.

Como o sinal de $|\det[D']|$ é função somente do seno de β , têm-se:

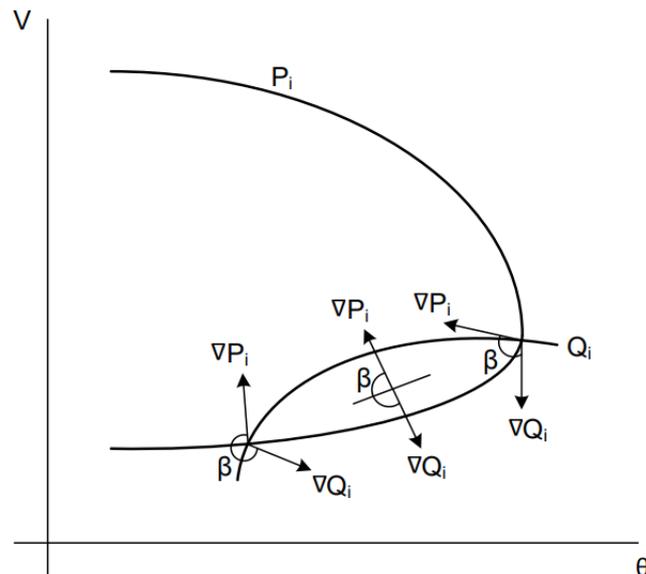
$|\det[D']| > 0$, se $\text{sen}(\beta) > 0$, isto é, $0^\circ < \beta < 180^\circ$

$|\det[D']| < 0$, se $\text{sen}(\beta) < 0$, isto é, $180^\circ < \beta < 0^\circ$

$|\det[D']| = 0$, se $\text{sen}(\beta) = 0$, isto é, $\beta \pm 180^\circ$, ou $\beta = 180^\circ$

Pela Figura 6, observa-se que para um ponto de operação normal na barra (operação estável), o ângulo β será sempre $0^\circ < \beta < 180^\circ$ e na região anormal de operação (operação instável), este sempre ocorre em $180^\circ < \beta < 0^\circ$. No ponto de máximo carregamento, os vetores gradientes estão alinhados, e, portanto, o valor de β é 180° e o módulo do determinante é nulo. Tais definições são mostradas graficamente na Figura 6.

Figura 6 - Localização dos vetores gradientes de P_i e Q_i no plano θV



Fonte: Cid, (2013).

Pelas características apresentadas, têm-se que o módulo do determinante da matriz D' está compreendido no intervalo $(-\infty, +\infty)$, onde para valores muito grandes, o ponto de operação possui um alto nível de segurança em tensão e para valores próximos a 0, a tensão na barra em análise é crítica e beira o colapso (PRADA et al., 2002).

Com o índice $\det[D']$, é possível o desenvolvimento de uma Rede Neural Artificial (RNA) integrante do submódulo “Sistema Inteligente” da Figura 1, para auxílio na computação e avaliação de segurança em tensão de um sistema elétrico, indicando o índice dado uma entrada de parâmetros de medição das barras do sistema. Os princípios de funcionamento de uma RNA, bem como seu treinamento e avaliação serão abordados a seguir.

2.3 Redes Neurais Artificiais

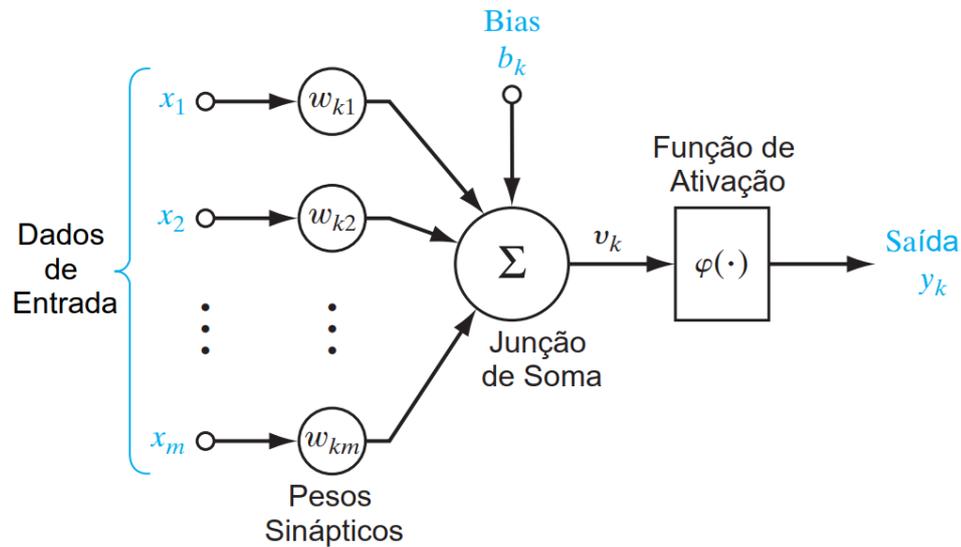
Uma Rede Neural Artificial é uma estrutura lógico-matemática que procura simular a forma, o comportamento e as funções de um neurônio biológico (FURTADO, 2019). Uma estrutura deste tipo, assim como o cérebro humano, possui a capacidade de reter conhecimento a partir de seu ambiente através de um processo de aprendizado, sendo este, armazenado entre as conexões neuronais conhecidas como pesos sinápticos (HAYKIN, 2009).

Nesta seção são apresentados os fundamentos e conceitos que regem o funcionamento de uma rede neural artificial, abordando a arquitetura da Rede Neural, o algoritmo de aprendizagem, e as formas de avaliar o desempenho final do modelo.

2.3.1 Princípios de Funcionamento e Arquitetura

Um neurônio é a unidade fundamental de uma RNA, e seu modelo mais utilizado é conhecido como Neurônio de McCulloch e Pitts (MCCULLOCH; PITTS, 1943), um neurônio com comportamento não-linear. Baseado no neurônio de McCulloch e Pitts, Rosenblatt (1958) propõe uma modelagem matemática que relaciona entrada e saída inspirada na biologia, o Perceptron. O modelo de um Perceptron pode ser visto na Figura 7.

Figura 7 – Representação do Perceptron de Rosenblatt.



Fonte: Traduzido de Haykin, (2009).

Nesta representação, identifica-se três elementos básicos:

1. Um conjunto de elos de conexão caracterizados por um sinal de entrada x_i associado a um peso sináptico w_i ;
2. Uma junção de soma para somar os dados de entrada, ponderados pela contribuição sináptica de cada neurônio constituindo um combinador linear;
3. Uma função de ativação para limitar a amplitude da saída y_k de um neurônio, limitando-o a uma faixa de amplitude permissível;

Além disto, o modelo de neurônio da Figura 7 também inclui uma unidade chamada *bias* (b_k), cuja função é aumentar ou diminuir a entrada líquida da função de ativação, dependendo de ser positiva ou negativa, respectivamente (HAYKIN, 2001).

Em termos matemáticos, um neurônio k pode ser descrito pelas seguintes equações:

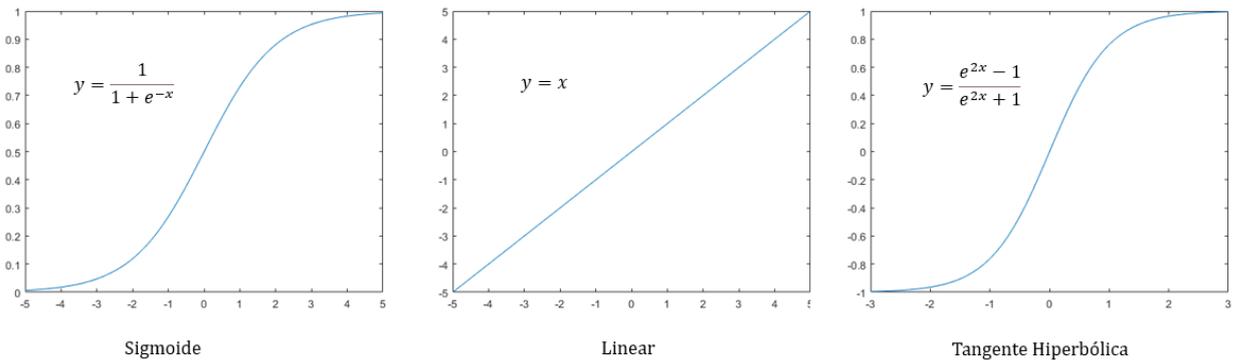
$$u_k = \sum_{i=1}^m w_{ki} x_i \quad (2.25)$$

E

$$y_k = \varphi(u_k + b_k) \quad (2.16)$$

As funções de ativação limitam a amplitude do sinal de saída do neurônio e a limitação normalmente se dá no intervalo fechado de $[0, 1]$ ou alternativamente em $[-1, 1]$, podendo também este intervalo estar entre $(-\infty, +\infty)$. Dentre muitas funções de ativação, a Figura 8 mostra as mais comuns.

Figura 8- Funções de Ativação

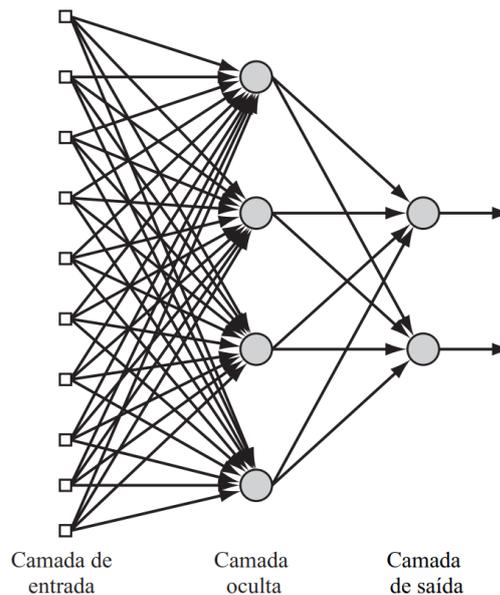


Fonte: Adaptado de HAYKIN, (2001) e FURTADO, (2019).

Em geral, somente um neurônio com muitas entradas como o da Figura 7 não é o suficiente para a realização de uma aplicação prática, desta forma, fazendo-se associações paralelas entre neurônios conectados a múltiplas camadas é possível ter uma rede capaz de se adaptar à aplicação proposta e obter uma boa generalização. Generalização em RNA's, segundo Haykin (2001) se refere à capacidade do sistema de produzir uma saída dada uma entrada não vista durante a etapa de treinamento. Além disso, uma RNA que contém neurônios não lineares pode ser aplicada em problemas cujo mapeamento entrada-saída é não linear (HAYKIN, 2001).

Dentre várias topologias de Redes Neurais, uma das mais conhecidas e aplicadas atualmente é a rede do tipo Rede de Propagação de Multicamadas (do inglês *Multilayer Feedforward Networks*), onde o fluxo de informação é unidirecional. Este tipo de rede apresenta uma ou mais camadas intermediárias (camadas ocultas) e uma camada de saída.

Figura 9 - Rede densa com uma camada oculta e uma camada de saída.



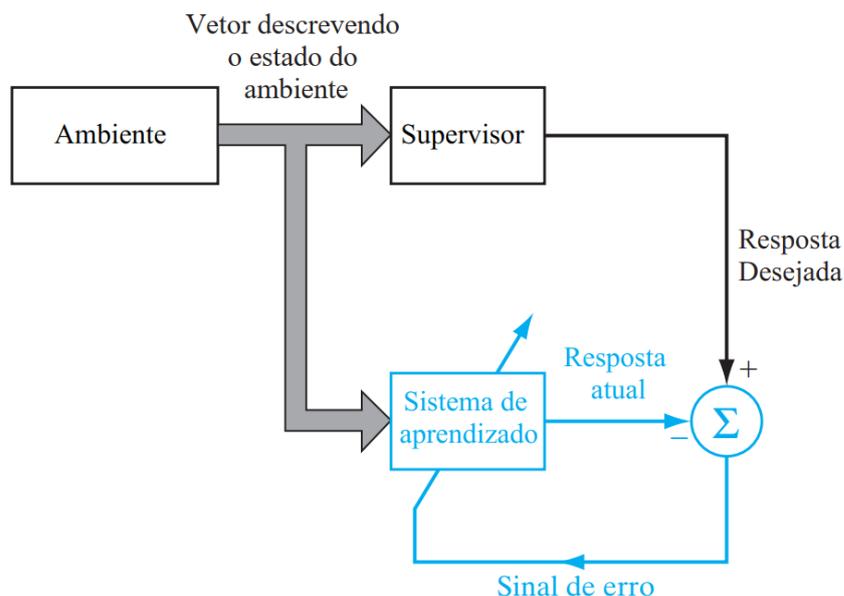
Fonte: Traduzido de Haykin, (2009).

2.3.2 Algoritmo de Aprendizagem

O conhecimento de uma RNA está contido nos pesos sinápticos, produto de um algoritmo computacional que segue uma sequência de eventos: o algoritmo de aprendizado. Os algoritmos de aprendizado dependem da arquitetura da rede neural, e podem ser divididos em três classes: aprendizado supervisionado, aprendizado não supervisionado e aprendizado por reforço (HAYKIN, 2001).

Para a arquitetura de rede neural proposta, usa-se algoritmos de aprendizado supervisionado que iterativamente permitem ao computador encontrar um padrão e informações ocultas entre os dados de entrada e saída, estabelecidos por um agente externo, o supervisor. O ajuste dos pesos é feito também pelo supervisor dado o erro obtido durante o fluxo direcional de informações na rede neural, ou seja, uma época de treinamento (HAYKIN, 2001). Tal conceito remete ao algoritmo de treinamento de Retropropagação (do inglês *Backpropagation*). Na Figura 10 está representado em diagrama de blocos, o treinamento supervisionado.

Figura 10 - Diagrama de blocos do algoritmo de aprendizado supervisionado.



Fonte: Traduzido de Haykin, (2009).

Desde que as funções de ativação sejam deriváveis, o método de retropropagação obtém uma boa generalização. Este método possui diversos parâmetros configuráveis que devem ser avaliados empiricamente para cada aplicação (HAYKIN, 2001):

- Inicialização dos pesos: É atribuído valores iniciais dos pesos dentro de uma faixa de valores, geralmente com valores de uma distribuição uniforme de média zero e variância escolhida;

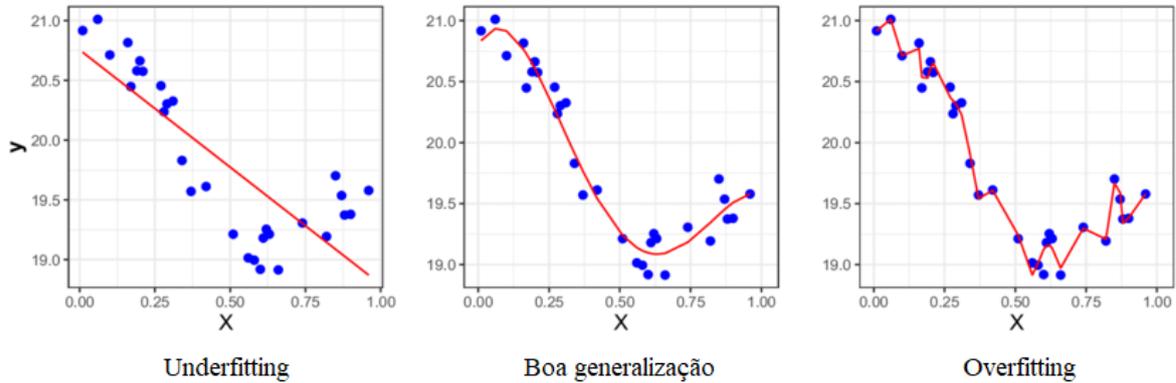
- Função de erro: Realizado ao fim de cada época, serve como parâmetro para avaliação de desempenho de treinamento da RNA, fundamental para os cálculos de otimização de pesos. Exemplos de funções de erro são o Erro Absoluto Médio e Erro Quadrático Médio.
- Otimizador: Responsável pelo cálculo de ajuste dos pesos da rede neural após propagação entrada-saída dos exemplos de treinamento. Este parâmetro procura uma solução global por meio de cálculos gradientes locais, no entanto, é possível que o algoritmo convirja em uma solução mínima local. Dentre muitos otimizadores, os mais utilizados dadas a topologia proposta são o método do Gradiente Descendente, RMSprop e Adam;
- Critério de parada: Pode ser feito atribuindo um limite máximo de épocas para o treinamento. Contudo, é desejável utilizar algoritmos de monitoramento do erro de treinamento ou validação do modelo (YING, 2019).

O treinamento da RNA é feito até que se tenha um bom conjunto de pesos para a aplicação proposta, ou seja, até que a RNA tenha um bom conhecimento adquirido, capaz de generalizar. Treinamentos por retropropagação podem se ajustar demais aos dados de treinamento, perdendo a capacidade de generalização caso não seja aplicadas técnicas de avaliação (YING, 2019).

2.3.3 Avaliação do modelo de RNA

O treinamento de uma RNA por retropropagação, segundo Haykin (2001) pode ter problemas de sobreajuste (do inglês *Overfitting*), onde a Rede Neural se ajusta demais aos dados de treinamento, perdendo a capacidade de generalização, e sub-ajuste (do inglês *Underfitting*), quando o modelo não é capaz de generalizar quando submetido a um exemplo de teste. Em outras palavras, um modelo com alta variância entre as previsões do modelo e valores esperados são características do sub-ajuste, e baixa variância do sobre-ajuste. Na Figura 11 é exemplificado por meio de gráficos de dispersão, dados de treinamento esperados, e a curva solução prevista pelo modelo de Rede Neural para cada caso.

Figura 11 – Exemplo de sistemas com *Overfitting*, *Underfitting*, e treinamento ótimo.

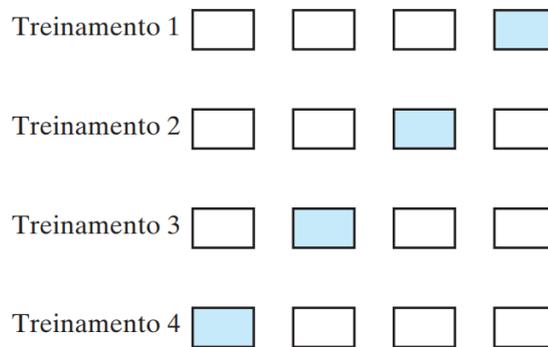


Fonte: Badillo et al., (2020).

A quantificação do sub-ajuste e sobreajuste deve ser feita por meio da Validação Cruzada (HAYKIN, 2001). A Validação Cruzada consiste em verificar a previsão da rede neural treinada com dados não vistos durante seu treinamento, que em outras palavras, quantifica a capacidade real de generalização do modelo (CHOI et al., 2020).

O método de Validação Cruzada, possui diversas variações, uma delas, o *K-fold*, que consiste em separar a base de dados em K subgrupos de dados de mesmo tamanho, após isso, realiza-se o treinamento do modelo adotando K-1 subgrupos como dados para treinamento, e um subgrupo para validação. No próximo treinamento-validação, faz-se o mesmo procedimento, porém utilizando outro subgrupo como dados de validação, e os demais como treinamento. O procedimento se repete até que K treinamentos de validação cruzada sejam realizados. A cada treinamento avalia-se o erro e variância do modelo. A Figura 12 ilustra o método de validação cruzada K-fold com $K = 4$.

Figura 12 - Ilustração do método de Validação Cruzada K-fold. Em azul o subgrupo de dados para validação, e em branco, para treinamento.

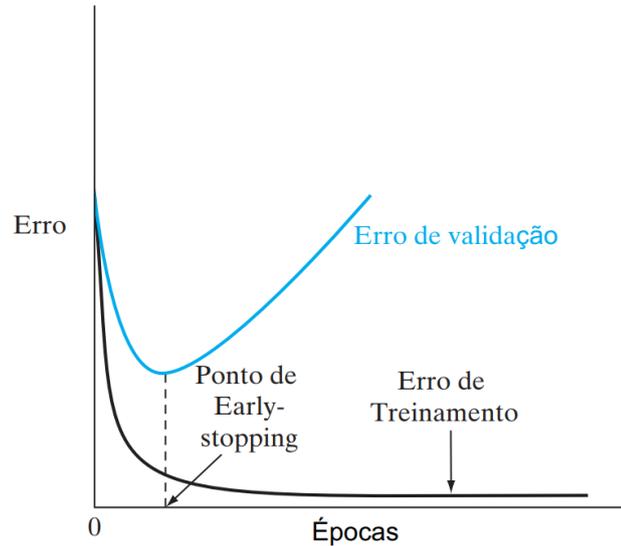


Fonte: Traduzido de HAYKIN, (2009).

O sobreajuste é bastante frequente em treinamentos de retropropagação. Segundo Koehrsen (2018), existem diversos métodos que atenuam esses efeitos, sendo um dos mais importantes, o *Early-stopping* (parada antecipada, em tradução literal), que deve ser ajustado

para interromper o treinamento quando o erro de validação aumenta substancialmente em relação ao erro de treinamento. A Figura 13 mostra um esboço das curvas de erro de treinamento e de validação em um sistema, nela percebe-se que em um dado momento o erro de validação é mínimo, antes de ascender.

Figura 13 - Curva de validação cruzada genérica em relação ao erro durante o treinamento.



Fonte: Traduzido de Haykin, (2009).

No ponto mínimo da curva do erro de validação da Figura 13, têm-se uma Rede Neural treinada com melhor desempenho, enquanto na região anterior ao ponto, sub-ajuste e na posterior, sobreajuste (HAYKIN, 2001; YING, 2019).

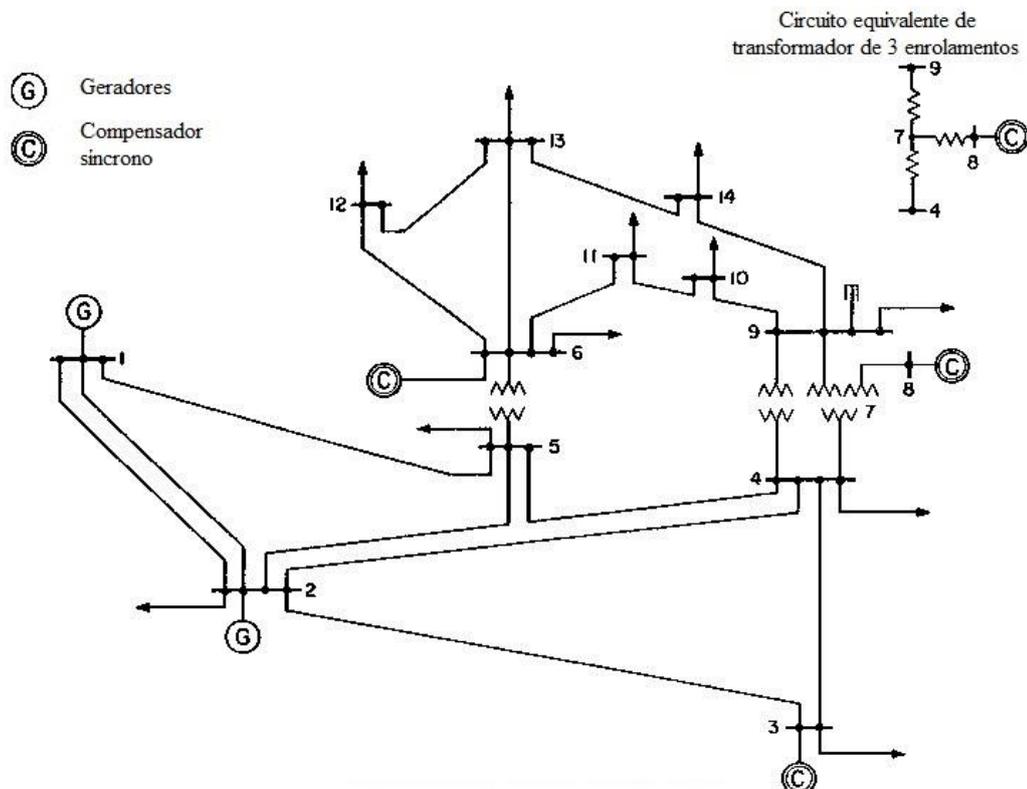
3 ESTUDO DE CASO

Neste capítulo, é apresentado primeiramente a descrição do sistema de teste utilizado para as análises de estabilidade de tensão e o *hardware* e *software* utilizados para a confecção do trabalho. Em seguida, é apresentado a metodologia de geração do banco de dados, que é constituído de parâmetros referentes ao sistema elétrico, e seus respectivos índices de estabilidade de tensão, também calculados. Por fim, faz-se o tratamento do banco de dados para a utilização no treinamento da RNA proposta e sua validação.

3.1 Descrição do sistema de testes IEEE 14-bus

O sistema IEEE 14 é amplamente utilizado em estudos baseados em fluxo de potência, para a análise de Estabilidade de Tensão (DHARMAPALA et al., 2020; QUEEN et al., 2021). Ele reproduz uma pequena parte do sistema elétrico dos Estados Unidos no ano de 1962 (“Power Systems Test Case Archive - UWEE”, [s.d.]). O sistema é composto por 14 barras interligadas por 17 linhas. Demais dados relativos ao sistema IEEE 14-bus podem ser consultados no ANEXO A.

Figura 14 - Sistema de testes IEEE 14-bus.



Fonte: “Power Systems Test Case Archive - UWEE”, ([s.d.]

Neste sistema, em seu fluxo de potência com valores originais, as tensões variam de 1,01 a 1,09 P.U., evidenciando um baixo carregamento ao sistema. Percebe-se também que a barra 1 (Referência) possui uma grande robustez de geração de potência ativa, bem como a barra 2 do tipo PV. Nas barras 3, 6 e 8 encontram-se compensadores síncronos para compensação de reativos e também, há três transformadores sendo um deles de três enrolamentos. O sistema possui 11 cargas ligadas às barras, totalizando uma demanda de 259,00 MW de potência ativa.

3.2 Descrição do *hardware* e *softwares* utilizados

Todas as ferramentas computacionais utilizadas neste trabalho foram executadas em um computador com CPU Intel[®] Core[™] i5-5200U CPU 2.20GHz, com 12GB de memória RAM e GPU Intel[®] HD Graphics 5500, e sistema operacional Microsoft Windows 10 Home.

Para simulações de sistemas elétricos de potência, será utilizado o ambiente de programação do Matlab versão 2016a, um *software* criado para a computação numérica e visualização de alto desempenho, oferecendo bibliotecas pré-definidas bastante sofisticadas, como solução de problemas de otimização, processamento de sinais, manipulações algébricas e até Redes Neurais Artificiais. Para simulação de fluxo de potência, visando um bom desempenho computacional e precisão, será utilizado neste trabalho o pacote de ferramentas do Matpower (ZIMMERMAN; MURILLO-SANCHEZ, [s.d.]), um pacote de licença livre que utiliza o Matlab como base. Com ele é possível resolver problemas de simulação e otimização de sistemas de energia em estado estacionário, como Fluxo de Potência, Fluxo de Potência Continuado, Fluxo de Potência Ótimo, etc.

Os algoritmos de pré-processamento do banco de dados, treinamento e avaliação de Redes Neurais, e visualização gráfica, foram desenvolvidos sob linguagem de programação Python, na versão 3.6.10, no ambiente de programação (IDE) Spyder, disponível no pacote Anaconda 3. As bibliotecas utilizadas durante o processamento de dados foram o Numpy, Pandas e Matplotlib. Para desenvolvimento das Redes Neurais Artificiais, foram utilizadas as bibliotecas do Scikit-learn e TensorFlow com a interface de programação de aplicativos (API) Keras.

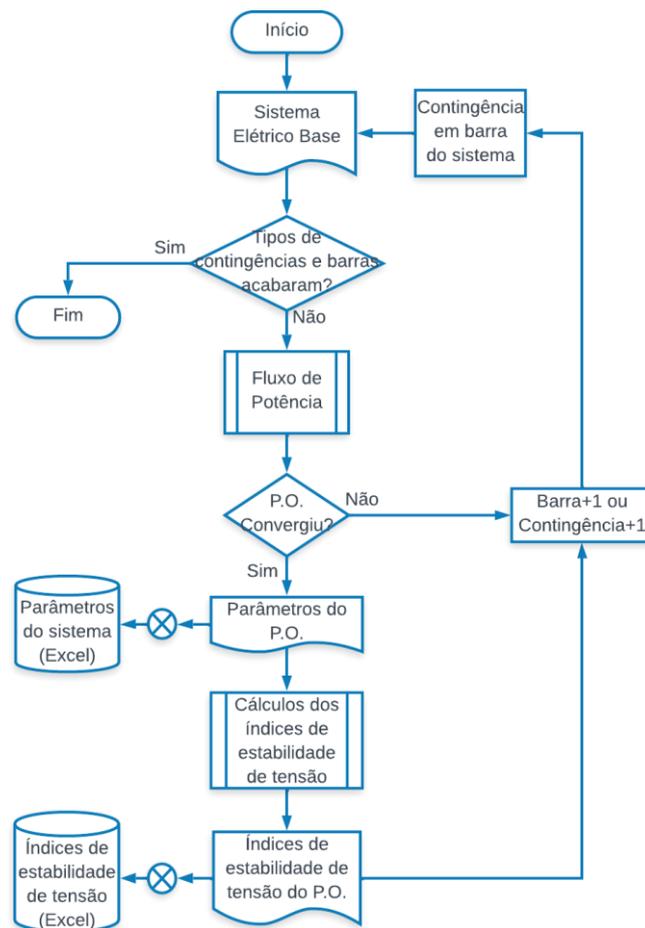
3.3 Geração do banco de dados para treinamento da RNA

Visando o treinamento da Rede Neural, propôs-se neste trabalho a geração de um conjunto de pontos operacionais, resultado de uma aplicação de contingências ao sistema elétrico IEEE 14-bus sem limites de geração às barras PV, com exportação em planilha Excel.

Além disso, a cada ponto de operação, e para cada barra do sistema, é calculado o índice de estabilidade de tensão ($\det[D']$), onde D' é definido na Equação 2.17. Para esse fim, foram utilizadas as ferramentas de manipulação de matrizes do *Software* Matlab e o pacote Matpower 7.1 para realização da modelagem e cálculo do fluxo de potência.

O fluxograma do método iterativo proposto obedece a sequência vista na Figura 15. Mais detalhes podem ser consultados no código fonte que se encontra no APÊNDICE B.

Figura 15- Fluxograma para gerar pontos de operação do sistema IEEE 14-bus.



Fonte: O autor.

Toda a lógica por trás da parada do algoritmo iterativo da Figura 15, bem como a forma de obtenção dos pontos operacionais com contingências e cálculo dos índices de estabilidade de tensão serão abordados a seguir.

3.3.1 Obtenção dos Pontos de Operação

O método de obtenção do ponto operativo da rede é baseado em fluxo de potência com método iterativo Newton Raphson, onde é encontrado de maneira otimizada no pacote Matpower.

O método iterativo visto na Figura 15 retorna ao usuário em formato de dados separados por vírgula, dados referentes ao módulo da tensão (V) em P.U. em cada barra do sistema, e seu respectivo ângulo (θ) em radianos, logo, a quantidade de colunas da matriz de dados retornada é 28, e a quantidade de linhas depende da quantidade de fluxos de potência realizados, o que depende das alterações de parâmetros aplicadas ao sistema de testes (contingências).

Foi proposto neste trabalho, a aplicação de três contingências ao sistema base IEEE 14-bus. As características da contingência, bem como os critérios para sua realização são abordados em detalhes nos tópicos a seguir.

3.3.1.1 Contingência 1

Esta contingência produz um aumento progressivo e gradual de Potência Ativa e Potência Reativa em todas as barras de carga (tipo PQ) do sistema elétrico com o seguinte incremento à cada fluxo de potência:

$$P = P_{base} \times \lambda \quad (3.1)$$

$$Q = Q_{base} \times \lambda \quad (3.2)$$

O fator multiplicador λ para essa contingência varia em passos de 0,3 a cada fluxo de potência realizado, tendo como valor inicial de 0,8.

A realização do aumento de carga à barra segue os critérios:

- Deve ser do tipo PQ somente;
- Deve possuir valores não nulos de potência ativa e reativa;
- O valor do incremento não deve ser maior que 6;

3.3.1.2 Contingência 2

Esta contingência se assemelha ao que é usualmente feito no método de fluxo de potência continuado, onde acrescenta-se um valor de potência à todas as barras que contém injeção de potência ativa ou reativa.

O aumento de carga nesta contingência obedece às equações 3.1 e 3.2, onde o fator multiplicador λ inicia-se em 0,5 e é incrementado em passos de 0,05 a cada novo fluxo de potência. Sua realização se aplica à todas as barras do sistema, não havendo incremento em barras que contenham potência ativa ou reativa nulas.

3.3.1.3 Contingência 3

Neste tipo de contingência, é proposto a geração de pontos operacionais com o sistema desprovido de uma de suas linhas de transmissão. Além disso, a carga do sistema é incrementada a cada iteração da mesma maneira vista na contingência 2.

Na contingência 3, a linha de transmissão a ser removida não pode estar ligada à barra de referência (tipo $V\theta$) do sistema, e conforme a contingência 2, o aumento de carga gradual e progressivo só terá efeito em barras que tenham algum tipo de carga não nula.

3.3.2 Cálculo do Índice de Estabilidade de Tensão $\det[D']$

Gerado o ponto de operação, obtém-se a matriz jacobiana correspondente à última iteração do fluxo de potência por Newton Raphson. O índice $\det[D']$ é calculado a cada barra do sistema, e para isso, separa-se para cada barra de análise, as derivadas parciais de P e Q em relação a V e θ , seguido do cálculo do determinante, como visto no Referencial Teórico.

Os resultados obtidos são armazenados na matriz de índices de estabilidade de tensão, esta que por sua vez apresenta 14 colunas referentes ao índice à cada barra do sistema, e quantidade de linhas a depender da quantidade de fluxos de potência realizados. Ao fim, exporta-se todos os índices de estabilidade de tensão para o formato de dados separados por vírgula, formando um segundo banco de dados.

3.3.3 Pré-processamento do Banco de Dados

O fluxograma de geração do banco de dados, como visto anteriormente, gera diferentes parâmetros do sistema em cada barra individual. No entanto, neste trabalho será utilizado somente os dados referentes as barras de carga, do tipo PQ, portanto o banco de dados de parâmetros do sistema será reduzido de 28 colunas para 18, bem como o dos índices de estabilidade de tensão de 14 para 9 colunas. Para tanto, será utilizado o pacote de ferramentas Pandas em linguagem de programação Python.

Como abordado em Haykin, (2009) para uma melhor eficiência no treinamento da rede neural, é interessante o uso de técnicas estatísticas para normalização dos dados de treinamento e teste, colocando-os em valores comuns.

Diante de várias opções vistas em Starovoitov e Golub (2021), foi utilizado a normalização linear descrita pela equação 3.3, que retorna valores no intervalo [0, 1]. A normalização será relativa à cada coluna do banco de dados, ou seja, a cada barra em análise do sistema elétrico.

$$X_{normalizado} = \frac{X - X_{min}}{X_{m\acute{a}x} - X_{min}} \quad (3.3)$$

3.4 Arquitetura da RNA

Como o objetivo da rede neural é a obtenção de um valor numérico correspondente aos índices de estabilidade de tensão de todas as barras, será utilizado uma rede do tipo Regressão, com múltiplas camadas com interligação do tipo densa.

A RNA proposta contará com quatro camadas, sendo duas camadas ocultas que darão à rede neural a capacidade de lidar com generalização em problemas não lineares (HAYKIN, 2001) como o Fluxo de Potência. Os neurônios das camadas ocultas terão unidades de bias unitários, e a quantidade de neurônios e função de ativação utilizados são:

- Primeira camada: Camada de entrada com 18 neurônios, referentes aos parâmetros V e θ de cada barra do tipo PQ, todos com função de ativação do tipo sigmoide;
- Segunda camada: Uma camada oculta cuja quantidade de neurônios é escolhida de maneira empírica. Para esta aplicação a segunda camada contará com 14 neurônios e função de ativação do tipo sigmoide;
- Terceira camada: É também uma camada oculta com 14 neurônios e função de ativação do tipo sigmoide;
- Quarta camada: Camada de saída referente à previsão dos índices de estabilidade de tensão da Rede Neural. Possui 9 neurônios correspondentes a cada barra do tipo PQ do sistema elétrico. Sua função de ativação é do tipo linear.

3.5 Treinamento da RNA

Para o treinamento da Rede Neural, foi utilizando o método *feedforward* por meio da API *Keras* em linguagem de programação Python. Para isso, os parâmetros de treinamento utilizados neste trabalho serão:

- Otimizador: Adam;
- Função de Erro: Erro Absoluto Médio;
- Épocas: 10000;
- Batch Size: 20 épocas.

Visando uma boa generalização da rede neural e prevenindo efeitos de Underfitting e Overfitting, o banco de dados previamente processado será dividido de forma aleatória em 80% para treinamento, e 20% validação, em ambos os dois bancos de dados.

Ao fim de cada época de treinamento (cada fluxo unidirecional dos dados) sobre a rede neural, o erro de treinamento é calculado e utilizado pelo otimizador no ajuste dos pesos da RNA. O erro de validação também é computado, porém utilizado para avaliação da capacidade de previsão da rede proposta, além de ser monitorado pela rotina *Early Stopping*, do pacote

Keras, que interromperá o treinamento antes de alcançar o número de épocas estabelecidos. Os parâmetros para a rotina *Early Stopping* são:

- Monitor: Erro de validação;
- Delta mínimo: 0,005;
- Paciência: 300 épocas;

A rotina de interrupção *Model Checkpoint*, também do pacote *Keras*, será implementada para salvar em disco todos os pesos do melhor modelo visto durante todo o processo de treinamento. Este por sua vez será utilizado para avaliação como modelo final deste trabalho.

3.6 Avaliação do modelo de RNA final

A Rede Neural obtida por meio do *Model Checkpoint* foi utilizada para avaliação, onde o erro de treinamento é obtido por meio da aplicação dos dados de treinamento e avaliação do erro obtido na previsão, e o erro de validação, consiste na aplicação de pontos operacionais não vistos durante o treinamento, para posterior computação do erro obtido. A base de dados de treinamento corresponde a 80% de todos os dados, enquanto de validação, 20%, e o erro computado na avaliação é o erro absoluto médio:

$$MAE = \frac{\sum_{i=1}^d |y_i - y'_i|}{d} \quad (3.4)$$

Onde y_i é o valor esperado na previsão, e y'_i o valor previsto pela RNA, e d o número total de dados de teste.

A aplicação do método *K-fold* de validação cruzada foi desenvolvida à parte considerando $K = 5$ (cada treinamento terá 80% de dados para treinamento e 20% para teste), considerando a mesma arquitetura de RNA salvo pelo *Model Checkpoint*. O critério de parada de treinamento em cada *fold* foi estabelecido pela rotina *Early Stopping* e o erro absoluto médio e variância de cada uma das *folds* foi avaliada.

4 RESULTADOS E DISCUSSÕES

O objetivo do sistema inteligente proposto é prever em barras de carga de um sistema elétrico o seu respectivo índice de estabilidade de tensão tendo como entrada o módulo e ângulo da tensão. Para a o treinamento e avaliação do modelo desenvolvido, foi utilizado bases de dados desenvolvidas de acordo com a metodologia proposta.

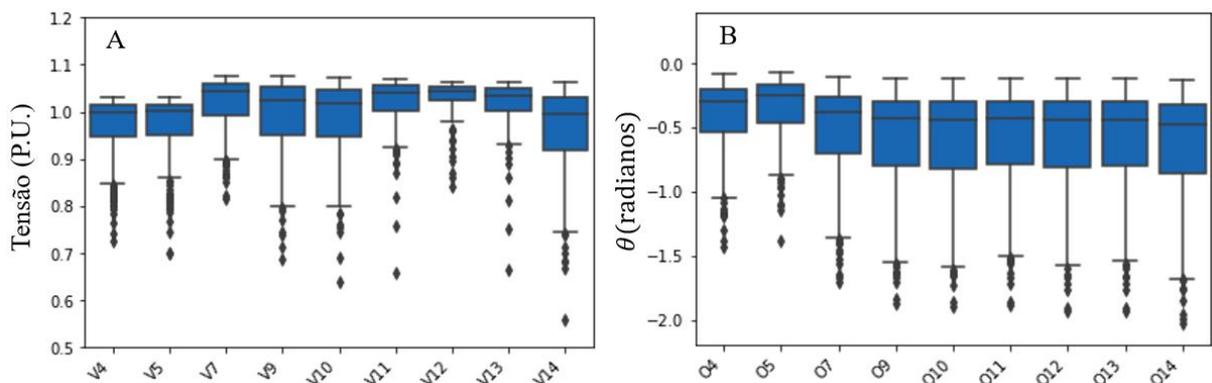
Nesta seção, são apresentadas as características do banco de dados gerado em Matlab com o auxílio do pacote Matpower, e avaliados os resultados do treinamento e teste do modelo de aprendizado de máquina, verificando a viabilidade de aplicação em um sistema de monitoramento *online*.

4.1 Banco de dados obtido pelo sistema IEEE 14-bus

De acordo com a metodologia acima descrita, foi obtido um total de 373 fluxos de potência convergidos, com um tempo computacional de 10,23 segundos. Destes, 144 foram produzidos durante a contingência 1, num tempo de 4,04 segundos, 71 para a contingência 2, com 1,79 segundos e 158 para a contingência 3, com 4,39 segundos de execução.

A matriz de parâmetros do sistema elétrico foi armazenada em uma planilha Excel, com dimensão de 373 linhas por 28 colunas, que por sua vez foram filtradas em Python, permanecendo apenas os dados referentes às barras de carga sem compensação síncrona, totalizando em uma matriz se 373 linhas por 14 colunas. Embora o banco de dados de parâmetros do sistema seja armazenado em um só documento, devido à escala dos dados, para melhor visualização é mostrado nas Figura 16a e 16b, gráficos do tipo boxplot, mostrando a distribuição dos módulos das tensões (em p.u.) e ângulo (em radianos) nas barras de carga.

Figura 16a - Gráfico boxplot dos módulos das tensões barras de carga; Figura 17b - Gráfico boxplot dos ângulos das tensões das barras de carga.



Fonte: O autor.

Observa-se que o conjunto de dados obtidos apresentou variados valores de módulos de tensão, onde a mediana correspondente a cada barra ficou compreendida entre 1,0 p.u. e 1,05

p.u., aproximadamente. Debaixo para cima, até o quartil número 1 mostra-se a distribuição de pouco mais de 25% dos dados de cada barramento (incluindo os pontos *Outliers*), no qual pode-se inferir que para cada barra do sistema foi obtido pontos de operação em que o nível de tensão pode ser considerado como crítico e à beira do colapso. Já para os ângulos das tensões em radianos, a mediana dos valores ficou próxima a -0,5, ângulo este próximo a -30°, referente ao ângulo da barra $V\theta$.

Na Figura 18, têm-se estatísticas numéricas e complementares à análise, referentes aos pontos operacionais obtidos, onde cada linha refere-se à uma barra do sistema elétrico IEEE14-bus. Em cada coluna, têm-se valores correspondentes a: valor médio, desvio padrão, valor máximo e mínimo, e os quartis 25%, 50% e 75%, e percebe-se que foi obtido uma gama de valores críticos de tensão a todos os barramentos, nos quais a barra com o nível mais baixo de tensão em um ponto operacional foi a barra número 14.

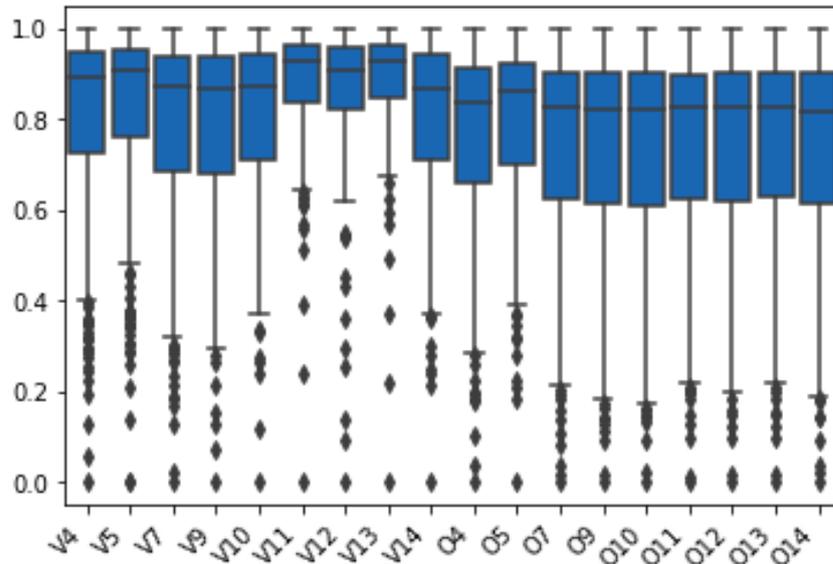
Figura 18 – Características estatísticas do banco de dados de parâmetros V e θ não normalizados.

Índice	Contagem	Média	Desvio P.	min.	25%	50%	75%	max
V4	373	0.971207	0.0588532	0.7241	0.94697	0.99714	1.014	1.0304
V5	373	0.972752	0.0618091	0.69853	0.95175	1.0002	1.0154	1.0311
V7	373	1.01938	0.0522375	0.81541	0.99305	1.0424	1.0589	1.075
V9	373	0.992276	0.0766463	0.68507	0.95096	1.0255	1.0524	1.0771
V10	373	0.988663	0.0740842	0.63932	0.94721	1.0176	1.047	1.0722
V11	373	1.02143	0.046893	0.65779	1.0017	1.0392	1.0549	1.0695
V12	373	1.0336	0.029491	0.83924	1.0237	1.0434	1.0542	1.0638
V13	373	1.01885	0.0423347	0.66384	1.0016	1.0331	1.0485	1.0624
V14	373	0.961065	0.0868301	0.55864	0.91683	0.99441	1.0319	1.062
O4	373	-0.405965	0.268274	-1.4354	-0.53822	-0.29922	-0.19814	-0.078027
O5	373	-0.34326	0.228016	-1.39	-0.4614	-0.24875	-0.16966	-0.065097
O7	373	-0.524758	0.340981	-1.7018	-0.70471	-0.38311	-0.25832	-0.10505
O9	373	-0.588803	0.375719	-1.8705	-0.79623	-0.43264	-0.29082	-0.11895
O10	373	-0.597643	0.381945	-1.9028	-0.81664	-0.43649	-0.29514	-0.11987
O11	373	-0.585319	0.374947	-1.892	-0.7834	-0.42397	-0.29439	-0.11608
O12	373	-0.596573	0.382385	-1.9364	-0.81487	-0.43429	-0.29821	-0.11713
O13	373	-0.599342	0.383317	-1.9376	-0.7973	-0.43536	-0.29917	-0.11827
O14	373	-0.638178	0.408933	-2.0334	-0.86214	-0.47433	-0.31419	-0.12708

Fonte: O autor

Na Figura 19 mostra-se a distribuição dos dados obtidos após a aplicação da normalização linear, evidenciando uma grande quantidade de pontos *Outliers*, ou seja, valores discrepantes que representam possíveis níveis críticos de tensão e ângulo.

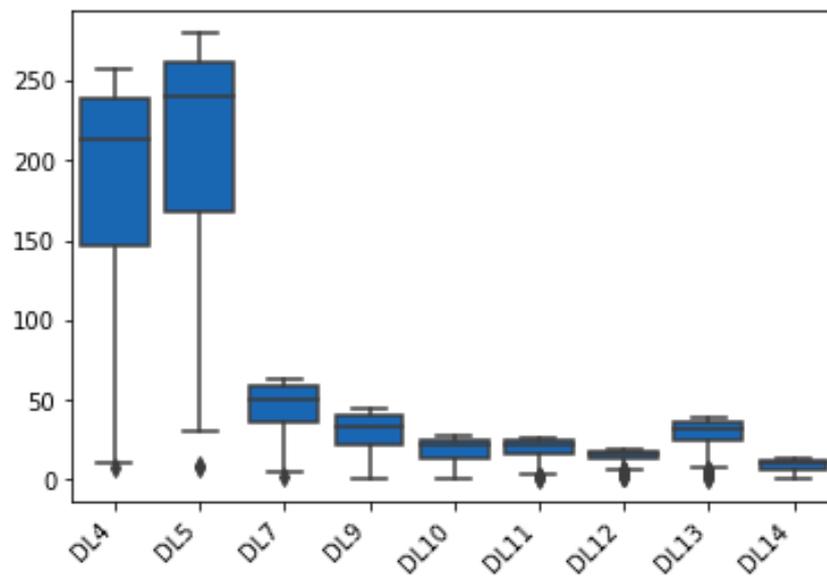
Figura 19 - Gráfico em Boxplot do banco de dados dos parâmetros do sistema após normalização.



Fonte: O autor.

Para cada fluxo de potência convergido, foi calculado o índice do determinante da Matriz D' a todas as barras do sistema, e armazenados em um outro arquivo Excel, cuja distribuição em gráfico de Boxplot pode ser vista na Figura 20.

Figura 20 - Gráfico em Boxplot do banco de dados dos índices de estabilidade de tensão.



Fonte: O autor.

Na Figura 21, têm-se estatísticas sobre os índices de estabilidade de tensão obtidos a cada barra, nos quais percebe-se um alto valor médio obtido nas barras 4 e 5, e valores mais críticos na barra 14.

Observa-se também a obtenção de pontos operacionais nos quais o nível estabilidade de tensão foi bastante crítico (próximo de zero).

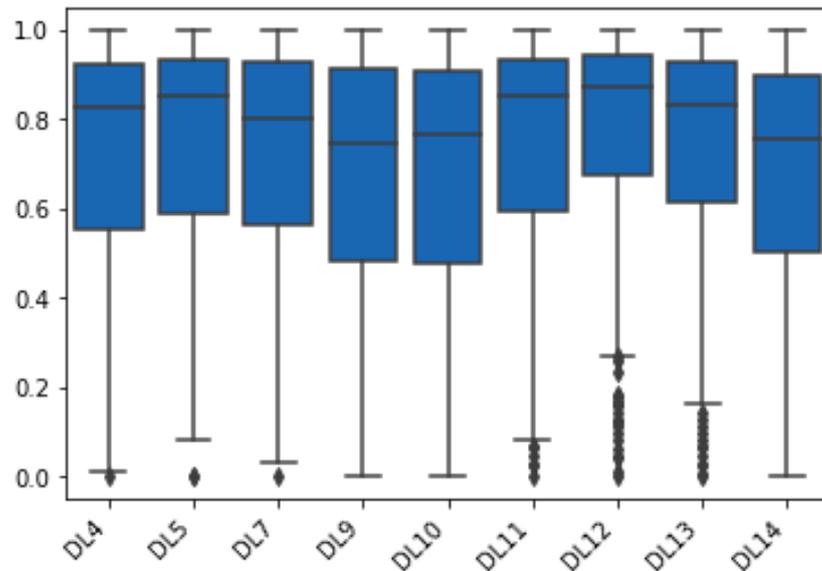
Figura 21 - Características estatísticas do banco de dados dos índices de estabilidade de tensão det[D'] não normalizados.

Índice	Contagem	Média	Desvio P.	min.	25%	50%	75%	max
DL4	373	187.744	61.4891	8.1223	146.55	214.08	238.53	257.39
DL5	373	208.758	65.7759	7.9574	168.04	240.02	261.48	280.36
DL7	373	45.9903	14.4146	2.4588	36.566	50.965	58.611	63.001
DL9	373	30.025	11.0897	1.2394	22.047	33.256	40.365	44.209
DL10	373	19.0683	7.23648	0.97123	13.857	21.659	25.455	27.911
DL11	373	19.5485	6.19817	0.73299	15.821	22.459	24.469	26.194
DL12	373	14.7463	4.01022	1.5529	13.113	16.54	17.739	18.712
DL13	373	29.037	8.64097	1.116	24.253	32.316	36.041	38.647
DL14	373	9.19424	3.46928	0.030742	6.9125	10.327	12.258	13.663

Fonte: O autor

O banco de dados de índices de estabilidade de tensão foi normalizado e todos os valores ficaram compreendidos no intervalo de dados [0, 1].

Figura 22 - Gráfico em Boxplot do banco de dados normalizados dos índices de estabilidade de tensão.



Fonte: O autor.

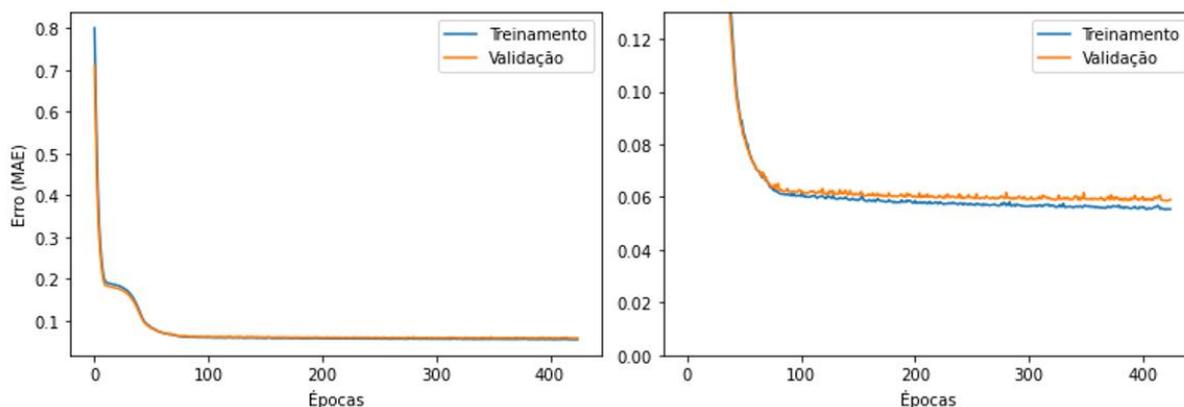
De acordo com a distribuição dos dados vistos na Figura 20, observa-se a robustez das barras de carga 4 e 5, com 50% dos índices de estabilidade de tensão concentrados em um intervalo de aproximadamente 150 a 250, principalmente por estarem próximas a fortes barras

de geração, porém, de acordo com as contingências administradas, foi obtido uma grande quantidade de índices relativamente críticos à ambas as barras. Além disso, as barras 11, 12 e 13, apresentam vários pontos *Outliers* próximos ao nível mínimo, que podem ser vistos com mais facilidade na Figura 22 tornando-as as barras mais críticas do sistema.

4.2 Treinamento da Rede Neural Artificial

Após o carregamento dos dados obtidos por fluxo de potência, foi realizado a filtragem e normalização os dados, que foram submetidos ao treinamento da RNA em linguagem Python. O treinamento foi interrompido na época 425 após 40,64 segundos pela rotina *Early Stopping*, e o modelo com menor erro de validação foi salvo pela rotina *Model Checkpoint* na época 423. A Figura 23 mostra em diferentes escalas, o erro de treinamento e o erro de validação, considerando a proporção de 80% do banco de dados para treinamento e 20% para validação.

Figura 23 - Erro absoluto médio de treinamento e validação em função das épocas de treinamento.



Fonte: O autor.

O treinamento é caracterizado por apresentar um forte erro nas primeiras épocas, explicado pela inicialização aleatória dos pesos sinápticos da RNA, que por meio dos otimizadores, é ajustado conforme uma quantidade de épocas é alcançada. O critério de parada antecipada da rotina *Early Stopping* é realizado quando o erro de validação não apresenta melhorias consideráveis, na época 425.

4.3 Avaliação do modelo final

Para que seja comprovada a eficiência e capacidade de generalização da RNA desenvolvida neste trabalho, foi utilizado o banco de dados de treinamento (80% dos dados) para previsão e verificação do erro de treinamento, e o banco de dados de teste (20% dos dados), que não foi visto pela RNA anteriormente. Os valores de erro absoluto médio (MAE) bem como o tempo decorrido podem ser vistos na Tabela 1.

Tabela 1 - Erro Absoluto médio para treinamento e teste do modelo final.

Tipo	MAE	Tempo de Execução do Teste (s)
Treinamento	0,05539	0,19107
Teste	0,05857	0,14893

Fonte: O autor

Utilizando a arquitetura proposta de RNA, aplicou-se o método K-Fold de validação cruzada, com $K = 5$, ou seja, em cada treinamento realizado, 80% dos dados foram utilizados para treinamento, e 20% para teste. A validação ocorreu intercalando cada subgrupo de dados para teste, e ao fim de cada treinamento, o erro absoluto médio considerando cada parcela foi computado. O tempo total de execução foi de 143,09 segundos e os valores de erro obtidos podem ser verificados na tabela abaixo.

Tabela 2 - Erro Absoluto médio para treinamento e teste do em validação cruzada K-fold.

Fold	Treinamento	Teste
1	0,0567	0,0544
2	0,0571	0,0572
3	0,0537	0,0619
4	0,0556	0,0595
5	0,0559	0,0509
MÉDIA	0,0558	0,0568

Fonte: O autor.

O resultado obtido apresentou uma média de erro de treinamento de 0,0558, enquanto na validação, 0,0568. O desvio padrão dos erros por sua vez foi de 0,0013 para o treinamento, e 0,0043 para validação, mostrando uma baixa variância e boa generalização do modelo obtido.

5 CONSIDERAÇÕES FINAIS

Neste trabalho foi desenvolvido uma metodologia para obtenção de pontos operacionais em um sistema elétrico de potência de testes, em conjunto com cálculos de índice de estabilidade de tensão, para criação e treinamento de uma Rede Neural Artificial capaz de atuar no monitoramento de segurança em tensão em tempo real em barras de carga. Para isto, foi proposto uma ampla revisão bibliográfica para entendimento da área de avaliação de segurança em um monitoramento *online* de SEP, além do estudo de Estabilidade de Tensão. Além disso, foi realizado uma avaliação dos mais relevantes estudos na área de previsão de índices de estabilidade de tensão com sistemas de aprendizado de máquina.

O problema de fluxo de potência é descrito com um conjunto de equações algébricas não lineares, onde a avaliação de segurança em tensão recorre a métodos computacionais de alto esforço computacional. Com isso, as redes neurais são recursos importantes para a abordagem do problema do controle preventivo, tendo em vista que, uma vez realizado o treinamento (atividade *off-line*), sua aplicação prática pode ser concluída com baixíssimo esforço computacional.

Os resultados obtidos neste trabalho mostram a eficiência da Rede Neural Artificial proposta para estimar devidamente os índices de estabilidade de tensão em barras do tipo PQ para casos não vistos durante seu treinamento, apresentando baixos níveis de erro de previsão. Além disso, o tempo gasto para a previsão da RNA treinada para múltiplas entradas foi evidenciado, mostrando um baixo esforço computacional e velocidade, permitindo assim que o operador do sistema tenha um bom indicativo da segurança em tensão que o ponto operacional em análise se encontra, possibilitando ações corretivas e evitando catástrofes. Contudo, a RNA proposta possui limitações, que devem ser complementadas e aprimoradas em trabalhos futuros.

5.1 Sugestões para trabalhos futuros

Nesta subseção são apresentadas algumas sugestões que visam dar continuidade à pesquisa iniciada neste trabalho de conclusão de curso:

- Avaliar a eficiência de redes neurais em sistemas de maior porte;
- Avaliar novas arquiteturas e métodos de treinamento de Redes Neurais para a aplicação proposta;
- Realizar o cálculo do índice de estabilidade de tensão $\det[D']$, não somente em barras de carga, mas também em barras de geração, barras com compensação síncrona e barras de referência, para aplicação em Redes Neurais Artificiais;

- Inserção de novas mudanças topológicas ao sistema elétrico de testes, como aumento de geração renovável próximo a carga (geração distribuída), curto-circuitos e limite de geração aos geradores do sistema, visando uma obtenção mais completa de dados;
- Utilização de outros métodos de fluxo de potência que contemplem pontos operacionais instáveis para treinamento da Rede Neural Artificial.

REFERÊNCIAS

- ADHIKARI, A. et al. **Comparison of voltage stability assessment using different machine learning algorithms**. 2020 IEEE 4th Conference on Energy Internet and Energy System Integration: Connecting the Grids Towards a Low-Carbon High-Efficiency Energy System, EI2 2020. **Anais...**Institute of Electrical and Electronics Engineers Inc., 30 out. 2020.
- ANDERSON, P. M.; FOUAD, A. A. **Power System Stability**. 2003.
- BADILLO, S. et al. An Introduction to Machine Learning. **Clinical Pharmacology & Therapeutics**, v. 107, 3 mar. 2020.
- CHOI, R. Y. et al. Introduction to Machine Learning, Neural Networks, and Deep Learning. **Translational vision science & technology**, v. 9, n. 2, p. 14, 2020.
- CID, P. H. S. Índices de Avaliação das Condições de Estabilidade de Tensão em Sistemas com Controle de Tensão Local, Remoto e Coordenado. **Rio de Janeiro**, 2013.
- DHARMAPALA, K. D. et al. Machine Learning Based Real-Time Monitoring of Long-Term Voltage Stability Using Voltage Stability Indices. **IEEE Access**, v. 8, p. 222544–222555, 2020.
- DURAI PANDY; INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. MADRAS SECTION; INSTITUTE OF ELECTRICAL AND ELECTRONICS ENGINEERS. **Proceedings of the 2017 IEEE International Conference on Intelligent Techniques in Control, Optimization & Signal Processing, INCOS'17 : 23rd-25th March 2017**. [s.l.: s.n.].
- FITIWI, D.; RAO, K. S. Neuro-Prony and Taguchi's Methodology-Based Adaptive Autoreclosure Scheme for Electric Transmission Systems. **Power Delivery, IEEE Transactions on**, v. 27, p. 575–582, 1 abr. 2012.
- FURTADO, M. I. V. **Redes neurais artificiais: uma abordagem para sala de aula**. **Ponta Grossa, PR. Atena Editora**, 2019.
- HATZIARGYRIOU, N. et al. Definition and Classification of Power System Stability – Revisited & Extended. **IEEE Transactions on Power Systems**, v. 36, n. 4, p. 3271–3281, 2021.
- HAYKIN, S. **Redes neurais: princípios e prática**. [s.l.: s.n.].
- HAYKIN, S. **Neural networks and learning machines, 3/E**. [s.l.] Pearson Education India, 2009.
- HUANG, G.-B.; ZHU, Q.-Y.; SIEW, C.-K. Extreme learning machine: Theory and applications. **Neurocomputing (Amsterdam)**, v. 70, n. 1, p. 489–501, 2006.
- KOEHRSEN, W. Overfitting vs. underfitting: A complete example. **Towards Data Science**, 2018.

- KUNDUR, P. **Power System Stability and Control**. McGraw-Hill, , 1994.
- KUNDUR, P. et al. Definition and classification of power system stability. **IEEE Transactions on Power Systems**, v. 19, n. 3, p. 1387–1401, ago. 2004.
- MCCULLOCH, W. S.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **The bulletin of mathematical biophysics**, v. 5, n. 4, p. 115–133, 1943.
- MONTICELLI, A. J. **Fluxo de carga em redes de energia elétrica**. [s.l.] E. Blucher, 1983.
- MORISON, K.; WANG, L.; KUNDUR, P. Power system security assessment. **IEEE power and energy magazine**, v. 2, n. 5, p. 30–39, 2004.
- OSEGI, E. N.; ANIREH, V. AMI: An auditory machine intelligence algorithm for predicting sensory-like data. **Computer Science**, v. 5, n. 2, p. 71–89, 2019.
- Power Systems Test Case Archive - UWEE**. Disponível em: <<https://labs.ece.uw.edu/pstca/>>. Acesso em: 25 nov. 2022.
- PRADA, R. B. et al. Voltage stability assessment for real-time operation. **IEE Proceedings-Generation, Transmission and Distribution**, v. 149, n. 2, p. 175–181, 2002a.
- PRADA, R. B. et al. Voltage stability assessment for real-time operation. **IEE Proceedings-Generation, Transmission and Distribution**, v. 149, n. 2, p. 175–181, 2002b.
- QUEEN, H. J. et al. Machine Learning-Based Predictive Techno-Economic Analysis of Power System. **IEEE Access**, v. 9, p. 123504–123516, 2021.
- ROSENBLATT, F. The perceptron: A probabilistic model for information storage and organization in the brain. **Psychological review**, v. 65, n. 6, p. 386–408, 1958.
- STAROVOITOV, V. V; GOLUB, YU. I. Data normalization in machine learning. **Informatika (Minsk, Belarus)**, v. 18, n. 3, p. 83–96, 2021.
- SUGANYADEVI, M. V.; BABULAL, C. K. Support Vector Regression Model for the prediction of Loadability Margin of a Power System. **Applied Soft Computing Journal**, v. 24, p. 304–315, 2014.
- USAMA, M. et al. **A mart oltage tability aneuver lgorithm oltage ollapses itigation**. [s.l: s.n.].
- VELOZA, O. P.; SANTAMARIA, F. Analysis of major blackouts from 2003 to 2015: Classification of incidents and review of main causes. **The Electricity journal**, v. 29, n. 7, p. 42–49, 2016a.
- VELOZA, O. P.; SANTAMARIA, F. Analysis of major blackouts from 2003 to 2015: Classification of incidents and review of main causes. **The Electricity journal**, v. 29, n. 7, p. 42–49, 2016b.

WOKOMA, B. A.; IDACHABA, A. O.; OSEGI, E. N. **Predicting Voltage Stability Indices of Nigerian 330kV 30 Bus Power Network Using an Auditory Machine Intelligence Technique; Predicting Voltage Stability Indices of Nigerian 330kV 30 Bus Power Network Using an Auditory Machine Intelligence Technique.** [s.l: s.n.]. Disponível em: <<http://yarpiz.com/323/ypml120-time-series->>.

YING, X. **An overview of overfitting and its solutions.** Journal of physics: Conference series. **Anais...**IOP Publishing, 2019.

ZAHEB, H. et al. A contemporary novel classification of voltage stability indices. **Applied Sciences**, v. 10, n. 5, p. 1639, 2020.

ZHANG, R. et al. **Voltage stability margin prediction by ensemble based extreme learning machine.** IEEE Power and Energy Society General Meeting. **Anais...**2013.

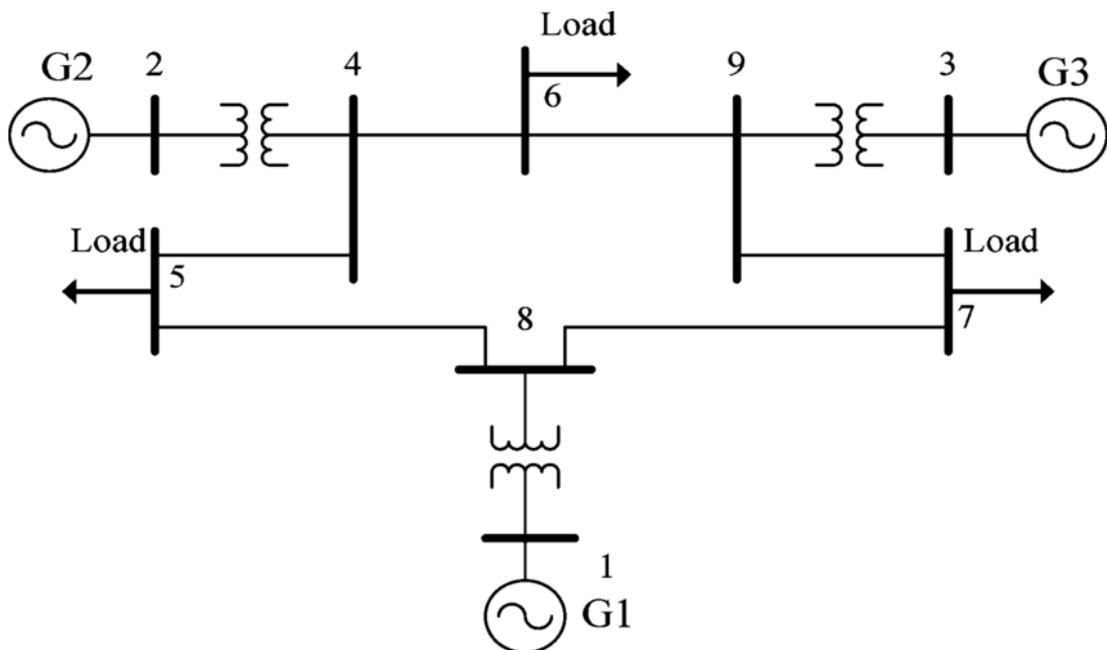
ZHOU, D. Q.; ANNAKAGE, U. D.; RAJAPAKSE, A. D. Online monitoring of voltage stability margin using an artificial neural network. **IEEE Transactions on Power Systems**, v. 25, n. 3, p. 1566–1574, ago. 2010.

ZIMMERMAN, R. D.; MURILLO-SANCHEZ, C. E. **MATPOWER (version 7.1)(2020).** , [s.d.].

APÊNDICE A – FLUXO DE POTÊNCIA

Um sistema elétrico de potência pode ser representado por diagramas unifilares com componentes que são classificados em dois grupos: os que estão ligados entre um nó qualquer e o nó-terra, como geradores, cargas, reatores e capacitores; e os que estão ligados entre dois nós quaisquer da rede, como é o caso das linhas de transmissão, transformadores e defasadores (MONTICELLI, 1983). Um exemplo de representação unifilar de um SEP de 9 barras pode ser visto na Figura 24.

Figura 24 - Representação unifilar do Sistema elétrico de potência IEEE 9-bus.



Fonte: (FITIWI; RAO, 2012).

Com isso, a partir do Fluxo de Potência (ou Fluxo de Carga), o estado operativo da rede pode ser obtido, como a distribuição dos fluxos e de algumas grandezas de interesse. Para isso faz-se uma modelagem estática dos componentes do sistema, com um conjunto de equações e inequações algébricas não lineares que precisam de um método numérico para sua resolução (MONTICELLI, 1983).

A.1 Modelagem de Fluxo de Potência

No problema de fluxo de carga, na formulação mais simples (formulação básica), associa-se quatro variáveis a cada barra do sistema (a cada nó), sendo que duas delas entram no problema como dados, e duas como incógnitas. A partir disto, classifica-se o tipo de barra conforme suas incógnitas, como vê-se na tabela abaixo.

Tabela 3 - Tipos de barras e incógnitas para a barra “i”.

Tipo da barra “i”	Dados	Incógnitas
PQ	P_i, Q_i	V_i, θ_i
PV	P_i, V_i	Q_i, θ_i
V θ (Referência)	V_i, θ_i	P_i, Q_i

As barras do tipo PQ são usadas para representar barras de carga, cujos dados de entrada P e Q são considerados constantes em relação aos níveis de tensão para toda resolução do problema. As barras PV são representações de pontos onde há geração ou algum tipo de compensação, neste caso a potência ativa gerada é constante, bem como o módulo da tensão. Já a barra V θ é a barra de referência que é única e exclusiva à um sistema e possui duas finalidades: servir como referência angular da tensão para o sistema (próprio nó terra); para fechamento do balanço de potência do sistema, levando em conta perdas de transmissão não conhecidas à priori (MONTICELLI, 1983).

Com base nas Leis de Kirchhoff, para cada barra são associadas duas equações, a primeira representa a injeção de potência ativa e a segunda, injeção de potência reativa, as quais estão representadas a seguir (MONTICELLI, 1983):

$$P_i = \sum_{m \in \Omega_i} P_{ij}(V_i, V_j, \theta_k, \theta_j) \quad (\text{A.2})$$

$$Q_i + Q_i^{sh}(V_i) = \sum_{m \in \Omega_i} Q_{ij}(V_i, V_j, \theta_i, \theta_j) \quad (\text{A.3})$$

Onde:

$i = 1, 2, \dots, \text{NB}$, sendo NB o número total de barras da rede;

Ω_i – Conjunto de barras vizinhas à barra i ;

V_i, V_j – Magnitudes das tensões das barras terminais do ramo $i - j$;

θ_i, θ_j – Ângulos das tensões das barras terminais do ramo $i - j$;

P_{ij} – Fluxo de potência ativa no ramo $i - j$;

Q_{ij} – Fluxo de potência reativa no ramo $i - j$;

Q_i^{sh} – Componente da injeção de potência reativa devida ao elemento *shunt* da barra k .

É necessário também para o problema de fluxo de potência, a limitação das amplitudes nodais das barras PQ, e de injeção de potência das barras PV:

$$V_i^{\min} \leq V_i \leq V_i^{\max} \quad (\text{A.4})$$

$$Q_i^{\min} \leq Q_i \leq Q_i^{\max} \quad (\text{A.5})$$

A.2 Solução do Problema Pelo Método de Newton Raphson

A complexidade na obtenção de uma solução para o fluxo de carga em um sistema de potência se dá pelas diferenças nos tipos de dados especificados para cada tipo de barra. Embora a formulação das equações não seja difícil, a forma de solução algébrica não é prática (MONTICELLI, 1983).

O método de Newton Raphson é amplamente utilizado para a solução do problema, e se baseia na expansão da série de Taylor para uma função de duas ou mais variáveis, onde para duas barras têm-se duas incógnitas e duas equações algébricas (MONTICELLI, 1983):

$$f_1(x_1, x_2) = K_1 \quad (\text{A.6})$$

$$f_2(x_1, x_2) = K_2 \quad (\text{A.7})$$

Estimando as soluções dessas equações como sendo $x_1^{(0)}$ e $x_2^{(0)}$, e $\Delta x_1^{(0)}$ e $\Delta x_2^{(0)}$ os valores somados a $x_1^{(0)}$ e $x_2^{(0)}$ para dar as soluções corretas, então pode-se escrever:

$$K_1 = f_1(x_1^{(0)}, x_2^{(0)}) = f_1(x_1^{(0)} + \Delta x_1^{(0)}, x_2^{(0)} + \Delta x_2^{(0)}) \quad (\text{A.8})$$

$$K_2 = f_2(x_1^{(0)}, x_2^{(0)}) = f_2(x_1^{(0)} + \Delta x_1^{(0)}, x_2^{(0)} + \Delta x_2^{(0)}) \quad (\text{A.9})$$

Resolvendo as equações 2.7 e 2.8 para $\Delta x_1^{(0)}$ e $\Delta x_2^{(0)}$ por série de Taylor:

$$K_1 = f_1(x_1^{(0)}, x_2^{(0)}) + \Delta x_1^{(0)} \left. \frac{\partial f_1}{\partial x_1} \right|_{(0)} + \Delta x_2^{(0)} \left. \frac{\partial f_1}{\partial x_2} \right|_{(0)} + \dots \quad (\text{A.9})$$

$$K_2 = f_2(x_1^{(0)}, x_2^{(0)}) + \Delta x_1^{(0)} \left. \frac{\partial f_2}{\partial x_1} \right|_{(0)} + \Delta x_2^{(0)} \left. \frac{\partial f_2}{\partial x_2} \right|_{(0)} + \dots \quad (\text{A.10})$$

Desprezando as derivadas parciais de ordem superior a 1, pode-se escrever as equações A.9 e A.10 de forma matricial conforme segue:

$$\begin{bmatrix} K_1 - f_1(x_1^{(0)}, x_2^{(0)}) \\ K_2 - f_2(x_1^{(0)}, x_2^{(0)}) \end{bmatrix} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} \begin{bmatrix} \Delta x_1^{(0)} \\ \Delta x_2^{(0)} \end{bmatrix} \quad (\text{A.11})$$

A matriz quadrada das derivadas parciais é conhecida como Matriz Jacobiana J, neste caso $J^{(0)}$.

$$J^{(0)} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \frac{\partial f_1}{\partial x_2} \\ \frac{\partial f_2}{\partial x_1} & \frac{\partial f_2}{\partial x_2} \end{bmatrix} \quad (\text{A.12})$$

Como $f_1(x_1^{(0)}, x_2^{(0)})$ é o valor calculado de K_1 dadas as estimativas iniciais $x_1^{(0)}$ e $x_2^{(0)}$, pode-se fazer:

$$\begin{bmatrix} \Delta K_1^{(0)} \\ \Delta K_2^{(0)} \end{bmatrix} = J^{(0)} \begin{bmatrix} \Delta x_1^{(0)} \\ \Delta x_2^{(0)} \end{bmatrix} \quad (\text{A.13})$$

Agora, aplicando a inversa do Jacobiano pode-se determinar $\Delta x_1^{(0)}$ e $\Delta x_2^{(0)}$. No entanto, como foi feito o truncamento na expansão da série, os valores calculados somados a estimativa inicial não determinam a solução do problema, o que se deve repetir o procedimento adotando novas estimativas $x_1^{(1)}$ e $x_2^{(1)}$, onde

$$x_1^{(1)} = x_1^{(0)} + \Delta x_1^{(0)} \quad (\text{A.14})$$

$$x_2^{(1)} = x_2^{(0)} + \Delta x_2^{(0)} \quad (\text{A.15})$$

O processo se repete até que as correções se tornem pequenas o suficiente para satisfazer uma precisão escolhida.

Para o problema de fluxo de carga com N barras, usa-se as tensões da barra i e da barra n, e admitâncias de linha in na forma polar, que resulta em:

$$P_i - jQ_i = \sum_{n=1}^N |V_i V_n Y_{in}| \angle (\theta_{in} + \delta_n - \delta_i) \quad (\text{A.16})$$

Então

$$P_i = \sum_{n=1}^N |V_i V_n Y_{in}| \cos(\theta_{in} + \delta_n - \delta_i) \quad (\text{A.17})$$

$$Q_i = - \sum_{n=1}^N |V_i V_n Y_{in}| \text{sen}(\theta_{in} + \delta_n - \delta_i) \quad (\text{A.18})$$

Nesse método, a barra referência do sistema deve ser omitida, pois o módulo e ângulo da tensão especificados são os próprios resultados do problema. As considerações sobre barras de tensão controlada serão abordadas mais à frente, e neste momento estima-se somente os valores de P e Q da barra do tipo PQ. Os valores P e Q especificados correspondem as constantes

K na equação A.11, e o módulo e ângulo da tensão, os valores estimados para $x_1^{(1)}$ e $x_2^{(1)}$. O processo é desenvolvido como visto anteriormente, até que se têm, através das equações A.18 e A.19:

$$\Delta P_i = P_{i,espec} - P_{i,calc} \quad (\text{A.18})$$

$$\Delta Q_i = Q_{i,espec} - Q_{i,calc} \quad (\text{A.19})$$

O processo iterativo é realizado até que os valores de ângulo e tensão satisfaçam uma precisão pré-estabelecida.

Em barras do tipo PV (tensão controlada), desde que o módulo de tensão seja constante, omite-se no Jacobiano a coluna de derivadas parciais com relação ao módulo da tensão da barra, e também as derivadas parciais em relação a Q, que podem ser determinados ao fim do processo iterativo de Newton Raphson.

APÊNDICE B – CÓDIGO FONTE DO SISTEMA DE GERAÇÃO DO BANCO DE DADOS

O código fonte para geração do banco de dados desenvolvido neste trabalho é dividido em dois arquivos: o MainFinal (do tipo script), e MTP (do tipo função).

B.1 Código fonte do arquivo MainFinal.m

Este é o script principal da aplicação, que tem a atribuição de escolher a contingência, o fator multiplicador, e a barra a ser incrementada (se houver) chamando a função MTP. A lógica também é responsável pela compilação de todos os pontos operacionais obtidos em uma única matriz, para exportação em arquivos de formato csv. O código fonte pode ser visto em linguagem Matlab abaixo.

```

%=====
% SCRIPT PRINCIPAL
% Administra contingências ao sistema elétrico base (IEEE14)
%=====
clc
clear

% Carregamento do sistema base IEEE14 do Matpower
%-----
IEEE14 = loadcase(case14);
% Número de barras e linhas de transmissão
NumLT = length(IEEE14.branch(:,1));
nb = length(IEEE14.bus(:,1));

% Inicialização de variáveis
%-----
Pf = 1;      % Contador de fluxos de potência
Flags = struct('success', 1, 'notPQ', 1);

% CONTINGÊNCIA 1: Incremento individual em barra do sistema IEEE14
% Informar em "i" a barra a ser incrementada
%-----
contingencia = 1;
tic
for i = 1:nb
    incremento = 0.8;
    while true,
        [IET, VO, Flags] = MTP(IEEE14, contingencia, i, incremento);
        if Flags.success ~= 1,
            % Não realizou o fluxo de potência pois o P.O. não
            % convergiu. Vai para a próxima barra.
            break
        elseif Flags.notPQ ~= 1,
            % Não realizou o fluxo de potência pois a barra "i" não é
            % do tipo PQ. Vai para a próxima barra
            break
        end
        incremento = incremento + 0.3;
        % Armazena dados do P.O. convergido
        Vbus(Pf,:) = (VO.V)';
    end
end

```

```

        ObusR(Pf,:) = (VO.TetaRad)';
        % Armazena índice det[D'] de todas as barras do P.O.
        DL2 (Pf,:) = (IET.detDL)';

        Pf = Pf+1;
    end
end
Tempo1 = toc;
Cont_Contingencial = Pf-1;
tic

% CONTINGÊNCIA 2: Incremento de todas as cargas em barras PQ do
% sistema base "IEEE14". Informar o incremento.
%-----
contingencia = 2;
incremento = 0.5;
while true,
    [IET, VO, Flags] = MTP(IEEE14, contingencia, i, incremento);
    if ~Flags.success,
        % Não realizou o fluxo de potência pois o P.O. não convergiu
        % Retorna para contingência 3
        break
    end
    incremento = incremento + 0.05;
    % Armazena dados do P.O. convergido
    Vbus(Pf,:) = (VO.V)';
    ObusR(Pf,:) = (VO.TetaRad)';
    % Armazena índice det[D'] de todas as barras do P.O.
    DL2 (Pf,:) = (IET.detDL)';

    Pf = Pf+1;
end
Tempo2 = toc;
Cont_Contingencial2 = (Pf-1) - Cont_Contingencial;

% CONTINGÊNCIA 3: Falta de linha de transmissão. Informar em "i" a
% linha a ser retirada de operação. Há também incremento a todas as
% barras do sistema
%-----
contingencia = 3;
tic
for i=1:NumLT,
    incremento = 1;
    while true,
        [IET, VO, Flags] = MTP(IEEE14, contingencia, i, incremento);
        if ~Flags.success,
            % Não realizou o fluxo de potência pois o P.O. não
            % convergiu. Vai para a próxima barra
            break
        end
        incremento = incremento + 0.3;
        % Armazena dados do P.O. convergido
        Vbus(Pf,:) = (VO.V)';
        ObusR(Pf,:) = (VO.TetaRad)';
        % Armazena índice det[D'] de todas as barras do P.O.
        DL2 (Pf,:) = (IET.detDL)';
    end
end

```

```

        Pf = Pf+1;
    end

end

Tempo3 = toc;
Pf = Pf-1;
Cont_Contingencia3 = Pf - Cont_Contingencia2;
VarInputs = [Vbus ObusR];

% Exporta parâmetros do sistema e índice em arquivo .csv
%-----
csvwrite('Inputs_VO.csv', VarInputs);
csvwrite('Output_DL2.csv', DL2);

```

B.2 Código fonte da função MTP

Script contendo uma função em Matlab, recebe como dados de entrada um sistema elétrico em formato estrutura, obtido na biblioteca do Matpower, a contingência escolhida, o índice do elemento único a ser incrementado (se necessário), e o multiplicador de carga λ . O retorno desta função é os parâmetros de Tensão e Ângulo em todas as barras para um dado ponto de operação e seus respectivos índices de estabilidade de tensão. A função também retorna sinalizadores ao script principal. O código fonte pode ser consultado abaixo.

```

%=====
% Função para aplicação de contingências, fluxo de potência e cálculo
% do índice de estabilidade de tensão
%=====
function [Indices2, VO, Flags] = MTP(IEEE14, contingencia, index, Lambda)
Indices2 = 1; VO = 1; Flags.success = 1; Flags.notPQ = 1;

% Verificação e aplicação de contingências ao sistema elétrico recebido
%-----
if contingencia == 1,
    % Análise de restrições
    if IEEE14.bus(index,2) ~= 1,
        Flags.notPQ = 0;
        return
    else
        if IEEE14.bus(index, [3 4])==0 | Lambda > 6,
            Flags.success = 0;
            return
        end
        % Aumento de carga em barra específica
        IEEE14.bus(index, [3 4]) = IEEE14.bus(index, [3 4]) * Lambda;
    end
elseif contingencia == 2,
    % Aumento de carga em todas as barras do sistema
    IEEE14.bus(:, [3 4]) = IEEE14.bus(:, [3 4]) * Lambda;
elseif contingencia == 3,
    % Análise de restrições
    if Lambda > 4,
        Flags.success = 0;
        return
    end
end

```

```

else
    % Desligamento da linha de transmissão "index"
    IEEE14.branch(index,11) = 0;
    % Aumento de carga em todas as barras do sistema
    IEEE14.bus(:, [3 4]) = IEEE14.bus(:, [3 4]) * Lambda;
end
end

% Retorna o resultado do fluxo de potência por Newton Raphson
%-----
results = runpf(IEEE14);
Flags.success = results.success;
if Flags.success ~= 1,
    %O fluxo de potência não convergiu
    return
end

% Armazena variáveis do ponto de operação
%-----
Vbus(1,:) = (results.bus(:,8))';
ObusR(1,:) = (results.bus(:,9))'*(pi/180);

% Gera matriz jacobiana completa
%-----
J = makeJac(results, 1);

% Atribuição de valores "Infinitos" a elementos de barras PV e V0 da
Jacobiana
%-----
kl=0;
nb = length(IEEE14.bus(:,1));
for i=1: nb,
    if IEEE14.bus(i,2) == 3,
        J(i,i)=10^10;
    end
    if IEEE14.bus(i,2) >= 2,
        J(i+nb,i+nb)=10^10;
        kl=kl+1;
    end
end

% Cálculo do índice det[D'] barra a barra
%-----
for BusA=1:nb
    %Obtenção das matrizes A, B, C e D
    A = J(1:BusA-1,1:end);
    A = [A;J(BusA+1:nb+BusA-1,1:end)];
    A = [A;J(nb+BusA+1:end,1:end)];
    B = A(1:end, BusA);
    B = [B A(1:end, nb+BusA)];
    T = A(1:end, 1:BusA-1);
    T = [T A(1:end, BusA+1:nb+BusA-1)];
    T = [T A(1:end, nb+BusA+1:end)];
    A = T;
    T = J(1:end, 1:BusA-1);
    T = [T J(1:end, BusA+1:nb+BusA-1)];
    T = [T J(1:end, nb+BusA+1:end)];
end

```

```

C = T (BusA, 1:end);
C = [C; T (nb+BusA, 1:end)];
D = J (BusA, BusA);
D = [D J (BusA, nb+BusA)];
T = J (nb+BusA, BusA);
T = [T J (nb+BusA, nb+BusA)];
D = [D;T];
DL = D-C*inv(A)*B;           % Obtenção da matriz D'
DL2 (BusA) = det (DL);      % Cálculo do índice det[D']
end

% Saida de dados
%-----
Indices2 = struct ('detDL', DL2);
VO = struct ('V', Vbus, 'TetaRad', ObusR);
end

```

APÊNDICE C – CÓDIGO FONTE DO ALGORITMO DE PRÉ-PROCESSAMENTO DO BANCO DE DADOS, TREINAMENTO DA REDE NEURAL E VALIDAÇÃO

O algoritmo foi desenvolvido em ambiente de programação (IDE) Spyder, utilizando a linguagem de programação Python. O código fonte é visto abaixo:

```

"""
Rede neural para avaliação de estabilidade de tensão em sistemas elétricos
"""

import pandas as pd
from sklearn.model_selection import train_test_split
from keras.models import Sequential, load_model
from keras.layers import Dense
from keras.callbacks import EarlyStopping, ModelCheckpoint
import time

# Funções
#-----
def criarRNA():
    RNA = Sequential() # classificador = nome da rede neural
    RNA.add(Dense(units=14, activation='sigmoid', input_dim=18,
                  bias_initializer='ones'))
    RNA.add(Dense(units=14, activation='sigmoid', bias_initializer='ones'))
    RNA.add(Dense(units=9, activation='linear'))
    RNA.compile(66revisões='adam', 66re='mae')
    return RNA

def norm(x, stats):
    Vmax = stats['max']
    Vmin = stats['min']
    normVal = (x - Vmin) / (Vmax-Vmin)
    flag_norm = True
    return (normVal, flag_norm)

# Callbacks
#-----
es = EarlyStopping(monitor='val_loss', mode='min', min_delta = 0.005,
                  verbose=1, patience = 300)
mc = ModelCheckpoint('best_model.h5', monitor='val_loss', mode='min',
                  verbose=1, save_best_only=True)

# Carregamento e tratamento do banco de dados
#-----
baseInput = pd.read_csv('Inputs_VOPQ.csv')
baseOutput = pd.read_csv('Output_DL2.csv')

barrasPQ = [4, 5, 7, 9, 10, 11, 12, 13, 14]
DelColIn = ['V1', 'V2', 'V3', 'V6', 'V8', 'O1', 'O2', 'O3', 'O6', 'O8']
DelColOut = ['DL1', 'DL2', 'DL3', 'DL6', 'DL8']

#Apaga colunas

```

```

baseInput = baseInput.loc[:, ~baseInput.columns.isin(DelColIn)]
baseOutput = baseOutput.loc[:, ~baseOutput.columns.isin(DelColOut)]

# Normalização Linear
classe_stats = baseOutput.describe().transpose()
previsores_stats = baseInput.describe().transpose()
baseOutputNorm, flag_norm = norm(baseOutput, classe_stats)
baseInputNorm, flag_norm = norm(baseInput, previsores_stats)

# Divisão do banco de dados em 80% para treinamento e #20% para teste
#-----
test_size = 0.2
prev_train, prev_test, classe_train, classe_test = train_test_split(
    baseInputNorm, baseOutputNorm, test_size=test_size)

# Criar Rede Neural
#-----
RNA = criarRNA()

batch_size = 20
epochs = 10000

Inicio = time.time()

# Treinar Rede Neural e obter dados do treinamento e validação
#-----
history = RNA.fit(prev_train, classe_train, validation_data=(
    prev_test, classe_test), batch_size=batch_size, epochs=epochs,
    67revisões67=[es, mc])

Fim = time.time()
TempoTrain = Fim - Inicio

# Carregament e avaliação do melhor modelo obtido durante o treinamento
#-----
saved_model = load_model('best_model.h5')

i = time.time()
train_acc = saved_model.evaluate(prev_train, classe_train, verbose=0)
f = time.time()
print('Tempo de Avaliação do Treinamento (s): %.5f'%(f-i))

# import tensorflow as tf

# mape = tf.keras.losses.MeanAbsolutePercentageError()
# mape(y_true, y_pred).numpy()

i = time.time()
test_acc = saved_model.evaluate(prev_test, classe_test, verbose=0)
f = time.time()
print('Tempo de Avaliação do Teste (s): %.5f'%(f-i))

print('Train: %.5f, Test: %.5f' % (train_acc, test_acc))

ev = []
ev.append([train_acc, test_acc])
67revisões_bm = saved_model.predict(prev_test)

```

```

# Plot da curva de aprendizado
#-----
# import matplotlib.pyplot as plt

# plt.figure(1)
# plt.xlabel('Épocas', fontsize=10)
# plt.ylabel('Erro (MAE)', fontsize=10)
# plt.plot(history.history['loss'], label='Treinamento')
# plt.plot(history.history['val_loss'], label='Validação')

# plt.legend()
# plt.show()

"""
#=====
#----- Validação Cruzada -----
#=====

from statistics import mean, stdev
from sklearn.model_selection import Kfold
Inicio = time.time()

skf = Kfold(n_splits=5, shuffle=True)
accu_train = []
accu_test = []

baseInputNormNP = baseInputNorm.to_numpy()
baseOuputNormNP = baseOuputNorm.to_numpy()

Inicio = time.time()

for train_index, test_index in skf.split(baseInputNormNP, baseOuputNormNP):
    x_train_fold, x_test_fold = baseInputNormNP[train_index],
    baseInputNormNP[test_index]
    y_train_fold, y_test_fold = baseOuputNormNP[train_index],
    baseOuputNormNP[test_index]
    RNA = criarRNA()
    RNA.fit(x_train_fold, y_train_fold, validation_data=(
        x_test_fold, y_test_fold), batch_size=batch_size, epochs=epochs,
        callbacks=[es, mc])

    accu_train.append(RNA.evaluate(x_train_fold, y_train_fold))
    accu_test.append(RNA.evaluate(x_test_fold, y_test_fold))

v = stdev(accu_train)
W = mean(accu_train)

Fim = time.time()
TempoCV = Fim - Inicio

"""
#=====

```

ANEXO A – DADOS DO SISTEMA IEEE 14-BUS

O sistema utilizado neste trabalho para avaliação de segurança de tensão e geração de banco de dados para treinamento de uma Rede Neural Artificial, foi o sistema de testes IEEE 14-bus, disponível na biblioteca Matpower, que por sua vez obteve da referência “Power Systems Test Case Archive - UWEE”, ([s.d.]).

Abaixo, o arquivo de dados do sistema IEEE 14-bus do Matpower.

```
function mpc = case14
%CASE14      Power flow data for IEEE 14 bus test case.
%   Please see CASEFORMAT for details on the case file format.
%   This data was converted from IEEE Common Data Format
%   (ieee14cdf.txt) on 15-Oct-2014 by cdf2matp, rev. 2393
%   See end of file for warnings generated during conversion.
%
%   Converted from IEEE CDF file from:
%       https://labs.ece.uw.edu/pstca/
%
%   08/19/93 UW ARCHIVE           100.0   1962 W IEEE 14 Bus Test Case
%
%   MATPOWER

%% MATPOWER Case Format : Version 2
mpc.version = '2';

%%----- Power Flow Data -----%%
%% system MVA base
mpc.baseMVA = 100;

%% bus data
%bus_i type   Pd   Qd   Gs   Bs   area   Vm   Va baseKV zone Vmax Vmin
mpc.bus = [
    1   3   0     0   0   0     1   1.06  0     0   1   1.06  0.94;
    2   2  21.7  12.7 0   0     1   1.045 -4.9   0   1   1.06  0.94;
    3   2  94.2  19   0   0     1   1.01  -12.00 0   1   1.06  0.94;
    4   1  47.8  -3.9 0   0     1   1.019 -10.00 0   1   1.06  0.94;
    5   1   7.6   1.6 0   0     1   1.02  -8.7   0   1   1.06  0.94;
    6   2  11.2   7.5 0   0     1   1.07  -14.00 0   1   1.06  0.94;
    7   1   0     0   0   0     1   1.062 -13.00 0   1   1.06  0.94;
    8   2   0     0   0   0     1   1.09  -13.00 0   1   1.06  0.94;
    9   1  29.5  16.6 0  19     1   1.056 -14.00 0   1   1.06  0.94;
   10   1   9     5.8 0   0     1   1.051 -15.00 0   1   1.06  0.94;
   11   1   3.5   1.8 0   0     1   1.057 -14.00 0   1   1.06  0.94;
   12   1   6.1   1.6 0   0     1   1.055 -15.00 0   1   1.06  0.94;
   13   1  13.5   5.8 0   0     1   1.05  -15.00 0   1   1.06  0.94;
   14   1  14.9   5   0   0     1   1.036 -16.00 0   1   1.06  0.94;
];

%% generator data
%   bus Pg      Qg      Qmax Qmin Vg  mBase status Pmax Pmin      Pc1
%   Pc2      Qc1min Qc1max Qc2min Qc2max ramp_agc      ramp_10
%   ramp_30 ramp_q  apf
mpc.gen = [
```

```

1      232.4  -16.9  10      0      1.06  100      1      332.4
0      0      0      0      0      0      0      0      0
0      0      0;
2      40     42.4  50     -40    1.045  100      1      140
0      0      0      0      0      0      0      0      0
0      0      0;
3      0      23.4  40     0      1.01   100      1      100
0      0      0      0      0      0      0      0      0
0      0      0;
6      0      12.2  24     -6     1.07   100      1      100
0      0      0      0      0      0      0      0      0
0      0      0;
8      0      17.4  24     -6     1.09   100      1      100
0      0      0      0      0      0      0      0      0
0      0      0;
];

%% branch data
%      fbus      tbus      r      x      b      rateA      rateB      rateC      ratio
%      angle      status      angmin      angmax
mpc.branch = [
1      2      0.01938 0.05917 0.0528 0      0      0      0      0
1      -360  360;
1      5      0.05403 0.22304 0.0492 0      0      0      0      0
1      -360  360;
2      3      0.04699 0.19797 0.0438 0      0      0      0      0
1      -360  360;
2      4      0.05811 0.17632 0.034  0      0      0      0      0
1      -360  360;
2      5      0.05695 0.17388 0.0346 0      0      0      0      0
1      -360  360;
3      4      0.06701 0.17103 0.0128 0      0      0      0      0
1      -360  360;
4      5      0.01335 0.04211 0      0      0      0      0      0
0      1      -360  360;
4      7      0      0.20912 0      0      0      0      0.978
0      1      -360  360;
4      9      0      0.55618 0      0      0      0      0.969
0      1      -360  360;
5      6      0      0.25202 0      0      0      0      0.932
0      1      -360  360;
6      11     0.09498 0.1989  0      0      0      0      0
0      1      -360  360;
6      12     0.12291 0.25581 0      0      0      0      0
0      1      -360  360;
6      13     0.06615 0.13027 0      0      0      0      0
0      1      -360  360;
7      8      0      0.17615 0      0      0      0      0
0      1      -360  360;
7      9      0      0.11001 0      0      0      0      0
0      1      -360  360;
9      10     0.03181 0.0845  0      0      0      0      0
0      1      -360  360;
9      14     0.12711 0.27038 0      0      0      0      0
0      1      -360  360;
10     11     0.08205 0.19207 0      0      0      0      0
0      1      -360  360;

```

```

    12    13    0.22092 0.19988 0    0    0    0    0
0      1    -360    360;
    13    14    0.17093 0.34802 0    0    0    0    0
0      1    -360    360;
];

%%----- OPF Data -----%%
%% generator cost data
%      1      startup shutdown      n      x1      y1      ...      xn
      yn
%      2      startup shutdown      n      c(n-1) ...      c0
mpc.gencost = [
    2      0      0      3      0.0430292599      20      0;
    2      0      0      3      0.25      20      0;
    2      0      0      3      0.01      40      0;
    2      0      0      3      0.01      40      0;
    2      0      0      3      0.01      40      0;
];

```