

**UNIVERSIDADE ESTADUAL PAULISTA “JÚLIO DE MESQUITA FILHO”
FACULDADE DE ENGENHARIA
CAMPUS DE ILHA SOLTEIRA**

André Carneiro Machado

Redes Neurais Artificiais Aplicadas na Análise de Fluxo de Potência

Ilha Solteira - SP
2025

TRABALHO DE GRADUAÇÃO
Curso de Graduação em Engenharia Elétrica

André Carneiro Machado

**REDES NEURAIS ARTIFICIAIS APLICADAS NA ANÁLISE DE FLUXO DE
POTÊNCIA**

Trabalho de Graduação, apresentado à Faculdade de Engenharia de Ilha Solteira – UNESP como parte dos requisitos para obtenção do título de Engenheiro Eletricista.

Nome do orientador

Prof. Dr. John Fredy Franco Baquero

Ilha Solteira - SP
2025

FICHA CATALOGRÁFICA
Desenvolvida pela Diretoria Técnica de Biblioteca e Documentação

M149r Machado, André Carneiro.
Redes neurais artificiais aplicadas em análise de fluxo de potência / André Carneiro Machado. -- Ilha Solteira: [s.n.], 2025
47 f. : il.

Trabalho de conclusão de curso (Graduação em Engenharia Elétrica) -
Universidade Estadual Paulista. Faculdade de Engenharia de Ilha Solteira, 2025

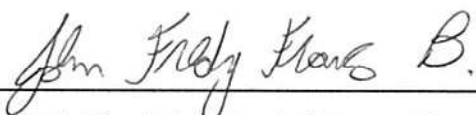
Orientador: John Fredy Franco Baquero
Inclui bibliografia

1. Aprendizado supervisionado. 2. Fluxo de potência. 3. Redes inteligentes.

Elaborada por Raiane da Silva Santos - CRB-8/9999

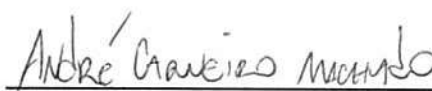
ATA DE DEFESA DE TRABALHO DE GRADUAÇÃO

Aos dezessete dias do mês de dezembro do ano de dois mil e vinte e cinco, o discente **André Carneiro Machado**, matriculado sob o nº182054691, tendo como banca examinadora o seu orientador, o *Prof. Dr. John Fredy Franco Baquero*, o Dr. Mario Andres Mejia Alzate e o *Doutorando Norberto Abrante Martinez*, apresentou o Trabalho de Graduação intitulado "**Redes Neurais Artificiais Aplicadas na Análise de Fluxo de Potência**", obtendo a nota 8,5 (oito vírgula cinco) e conceito APROVADO.



Prof. Dr. John Fredy Franco Baquero

- Orientador-



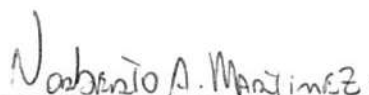
André Carneiro Machado

- Discente -



Dr. Mario Andres Mejia Alzate

- Membro da Banca -



Doutorando Norberto Abrante Martinez

- Membro da Banca -



Documento assinado digitalmente

MARIO ANDRES MEJIA ALZATE

Data: 17/12/2025 15:30:23-0300

Verifique em <https://validar.iti.gov.br>

AGRADECIMENTOS

Agradeço, primeiramente, à minha esposa Simone Suemi Iseri, pelo amor, compreensão e apoio incondicional em todas as etapas desta jornada. Sua presença constante foi essencial para que eu mantivesse o equilíbrio e a motivação necessários para concluir este trabalho.

Agradeço também ao meu orientador Prof. Dr. John Fredy Franco Baquero, pela orientação precisa, paciência e dedicação ao longo de todo o processo de pesquisa e desenvolvimento. Sua experiência e rigor científico foram fundamentais para o amadurecimento deste projeto e para minha formação como pesquisador.

Estendo meus agradecimentos à Universidade Estadual Paulista “Júlio de Mesquita Filho” – UNESP, campus de Ilha Solteira, pelo ambiente acadêmico que possibilitou o desenvolvimento deste trabalho, e a todos os professores e colegas que contribuíram, direta ou indiretamente, para o aprendizado e crescimento intelectual alcançados durante esta trajetória.

Por fim, agradeço a todos que, de alguma forma, acreditaram neste projeto e ofereceram apoio, incentivo ou inspiração. A cada um, minha sincera gratidão.

O sonho dos homens não tem fim.
— Marshall D. Teach, *One Piece*

RESUMO

Este trabalho investiga a aplicação de Redes Neurais Artificiais (RNAs) para a análise de fluxo de potência em sistemas elétricos, propondo um pipeline híbrido que integra modelagem física clássica e aprendizado supervisionado. A etapa de aprendizado de máquina utiliza Random Forest e XGBoost para determinar a importância das variáveis e orientar a engenharia de features, culminando no treinamento de uma RNA do tipo Perceptron Multicamadas (MLP) implementada em TensorFlow/Keras. O método permite comparar o desempenho da RNA em relação aos métodos clássicos Newton-Raphson e Gauss-Seidel, considerando precisão, robustez e tempo computacional. O estudo adota o sistema IEEE de 14 barras como caso-base, modelado no Pandapower e alimentado por dados reais da curva de carga horária da região Sudeste (ONS, 2023). Os resultados demonstram que a RNA reproduz o comportamento do fluxo de potência de referência com $R^2 \approx 0,995$ para ângulos de tensão e $R^2 \approx 0,955$ para magnitudes, alcançando uma redução superior a 99 % no tempo de inferência. Conclui-se que o uso de RNAs na estimativa de variáveis elétricas é uma alternativa viável e eficiente, com potencial para aplicações em sistemas elétricos inteligentes e operações em tempo real.

Palavras-chave: Aprendizado supervisionado. Fluxo de potência. Pandapower. Redes neurais artificiais. Redes Inteligentes.

ABSTRACT

This work investigates the application of Artificial Neural Networks (ANNs) to power flow analysis in electrical systems, proposing a hybrid pipeline that integrates classical physical modeling with supervised learning techniques. The machine learning stage employed Random Forest and XGBoost to evaluate feature importance and guide feature engineering, followed by training of a Multilayer Perceptron (MLP) implemented in TensorFlow/Keras. The methodology enables direct comparison of ANN performance with classical methods such as Newton-Raphson and Gauss-Seidel, assessing accuracy, robustness, and computational efficiency. The IEEE 14-bus system was adopted as the test case, modeled using Pandapower and fed with real data from the hourly load curve of the Southeast region of Brazil (ONS, 2023). The ANN reproduces the reference power-flow behavior, with $R^2 \approx 0.995$ for voltage angles and $R^2 \approx 0.955$ for voltage magnitudes, while reducing inference time by more than 99 %. The results confirm the feasibility and effectiveness of ANNs for real-time estimation of electrical variables, highlighting their potential in smart grid environments and intelligent power-system operations.

Keywords: Artificial neural networks. Pandapower. Power flow. Smart grids. Supervised learning.

LISTA DE FIGURAS

Figura 1	- Representação de um neurônio biológico.....	19
Figura 2	- Perceptron multicamadas.....	20
Figura 3	- Arquiteturas de RNAs.....	23
Figura 4	- Média de carga por hora do dia.....	28
Figura 5	- Média de carga por dia da semana.....	28
Figura 6	- Média de carga mês-a-mês.....	28
Figura 7	- Heatmap (MAE) para arquiteturas.....	35
Figura 8	- MAE médio de Vm (p.u.)	35
Figura 9	- MAE médio de VA (graus).....	36
Figura 10	- Previsto x Real de VM (p.u.)	38
Figura 11	- Previsto x Real de VA (graus).....	38
Figura 12	- MAE médio por barra (VM)	39
Figura 13	- MAE médio por barra (VA)	39
Figura 14	- Gráfico de importância de variáveis.....	41

LISTA DE TABELAS

Tabela 1	- Funções de ativação.....	21
Tabela 2	- Resultados Grid Search Vm.....	33
Tabela 3	- Resultados Grid Search VA.....	34
Tabela 4	- Importância das Features.....	40

LISTA DE SIGLAS E ABREVIATURAS

AC – *Alternating Current* (Corrente Alternada)

ANN – *Artificial Neural Network* (Rede Neural Artificial)

API – *Application Programming Interface* (Interface de Programação de Aplicações)

CSV – *Comma-Separated Values* (Valores Separados por Vírgulas)

DC – *Direct Current* (Corrente Contínua)

GNN – *Graph Neural Network* (Rede Neural em Grafos)

GPU – *Graphics Processing Unit* (Unidade de Processamento Gráfico)

IEEE – *Institute of Electrical and Electronics Engineers*

K-Fold – *K-Fold Cross-Validation* (Validação Cruzada em K Partições)

Keras – *API* de alto nível para aprendizado profundo integrada ao TensorFlow

MAE – *Mean Absolute Error* (Erro Absoluto Médio)

ML – *Machine Learning* (Aprendizado de Máquina)

MLP – *Multilayer Perceptron* (Perceptron Multicamadas)

ONS – Operador Nacional do Sistema Elétrico

PMU – *Phasor Measurement Unit* (Unidade de Medição Fasorial)

Pu – *Per Unit System* (Sistema por Unidade)

RNA – Rede Neural Artificial

RMSE – *Root Mean Square Error* (Raiz do Erro Quadrático Médio)

R^2 – *Coefficient of Determination* (Coeficiente de Determinação)

SEP – Sistema Elétrico de Potência

SIN – Sistema Interligado Nacional

TCC – Trabalho de Conclusão de Curso

TensorFlow – Framework para aprendizado profundo

XGBoost – *Extreme Gradient Boosting* (Técnica de Conjunto de Árvores com Gradiente Extremo)

LISTA DE SÍMBOLOS

$\Delta P, \Delta Q$ – Vetores de desbalanço de potência ativa e reativa (MW, MVar)

$\Delta\theta, \Delta V$ – Correções de ângulo e tensão nas iterações do método Newton-Raphson (graus, p.u.)

ε – Tolerância de convergência do método numérico

λ – Parâmetro de regularização (em XGBoost e modelos de regressão).

θ_i – Ângulo de fase da tensão na barra i (graus)

$\theta_{ij} = \theta_i - \theta_j$ – Diferença angular entre as tensões das barras i e j (graus)

η – Taxa de aprendizado (learning rate)

B_{ij} – Susceptância entre as barras i e j

b – Bias (termo de deslocamento) em neurônios artificiais

E – Função de erro ou perda (loss function)

$f(t)$ – Fator de carga normalizado no tempo t (adimensional)

$f(v)$ – Função de ativação aplicada à soma ponderada

G_{ij} – Condutância entre as barras i e j

J – Matriz jacobiana do sistema (adimensional)

MAE – Mean Absolute Error (Erro Médio Absoluto)

n – Número total de amostras do conjunto de dados

P_i – Potência ativa injetada na barra i (MW)

Q_i – Potência reativa injetada na barra i (MVar)

R^2 – Coeficiente de determinação (adimensional)

RMSE – Root Mean Square Error (Raiz do Erro Quadrático Médio)

t_i – Instante de tempo correspondente à amostra i (h)

V_i – Módulo da tensão na barra i (p.u.)

V_j – Módulo da tensão na barra j (p.u.)

w_i – Peso sináptico associado à entrada x_i

x_i – Variável de entrada do neurônio (normalizada)

y – Saída do neurônio (valor previsto)

Sumário

1	INTRODUÇÃO	14
1.1	Justificativa	15
1.2	Objetivos.....	15
2	REVISÃO DE LITERATURA	17
2.1	Fluxo de Potência.....	17
2.1.1	Newton-Raphson: Formulação Matricial e Pseudocódigo	17
2.1.2	Gauss-Seidel: Algoritmo Iterativo e Convergência	18
2.1.3	Comparação de Desempenho	18
2.2	Redes Neurais Artificiais	18
2.2.1	O Neurônio Artificial: Inspiração Biológica e Modelo Matemático.....	19
2.2.2	Funcionamento de um Neurônio Artificial.....	19
2.2.3	Componentes essenciais de uma RNA	20
2.2.4	Paradigmas de Aprendizado de Máquina para RNAs.....	21
2.2.5	Treinamento de RNAs: O Algoritmo Backpropagation.....	21
2.2.6	Arquiteturas de RNAs Relevantes para Engenharia	22
2.3	Redes Neurais Aplicadas em Modelagem de Sistemas Elétricos.....	24
2.3.1	Estimação de estado e aprendizado supervisionado	24
2.3.2	Modelagem do fluxo de potência	24
2.3.3	Hibridização com algoritmos clássicos e aprendizado de características	24
2.3.4	Aplicações emergentes e perspectivas	24
2.4	Bibliotecas em Python	25
2.4.1	Pandapower — Modelagem e Simulação Elétrica	25
2.4.2	Pandas — Estruturação e Manipulação de Dados	25
2.4.3	Processamento Numérico Vetorial	25
2.4.4	Aprendizado Supervisionado e Pré-Processamento	26
2.4.5	TensorFlow e Keras — Modelagem Neural Profunda	26
2.4.6	XGBoost — Ensemble Gradient Boosting.....	26
2.4.7	Matplotlib e Seaborn — Visualização e Análise Gráfica.....	26
3	MÉTODO PROPOSTO PARA ANÁLISE DE FLUXO DE POTÊNCIA USANDO REDES NEURAIS	27
3.1	Curva de Carga	27
3.1.1	Obtenção dos dados.....	27
3.1.2	Filtragem Para a Região Sudeste	27
3.1.3	Geração dos Gráficos Estatísticos.....	27

3.1.4	Utilização dos Dados	29
3.2	Modelagem Elétrica	29
3.2.1	Geração de Cenários Físicos	29
3.2.2	Pré-processamento e Normalização	29
3.3	Modelos de Aprendizado Supervisionado	30
3.3.1	Random Forest	30
3.3.2	XGBoost	30
3.3.3	Modelagem Neural (RNA – MLP)	30
3.3.4	Validação	31
3.4	Grid Search.....	31
3.4.1	Espaço de Busca	32
4	RESULTADOS E DISCUSSÃO	33
4.1	Grid Search.....	33
4.1.1	Sumário dos desempenhos	33
4.1.2	Implicações e recomendações	36
4.2	Desempenho da RNA.....	37
4.2.1	Avaliação.....	37
4.3	Importância das Variáveis.....	40
4.4	Robustez e Generalização.....	41
4.5	Discussão	42
5.	CONCLUSÃO	44
6.	Referências.....	46

1 INTRODUÇÃO

A análise de fluxo de potência constitui um dos pilares da Engenharia Elétrica, sendo fundamental para o planejamento, operação e expansão confiável das redes elétricas de transmissão e distribuição. As primeiras técnicas de análise na história remontam à década de 1950, quando métodos como Gauss-Seidel e Newton-Raphson foram adaptados para ambientes computacionais emergentes. Tais algoritmos, baseados em iterações sucessivas, permitiam pela primeira vez a resolução eficiente de sistemas não lineares de grande porte, viabilizando o dimensionamento adequado de linhas, transformadores e barras de geração de carga (STOTT & ALSAC, 1974).

Historicamente, a evolução dos métodos de fluxo de potência acompanha o progresso tecnológico, a expansão da distribuição de energia e a crescente complexidade das redes elétricas. Inicialmente, o foco das análises eram sistemas radiais e malhados sob condições estáticas, com hipóteses simplificadoras quanto ao comportamento das cargas e geradores. Com a criação do computador digital, tornou-se possível abordar sistemas mais complexos, incorporando modelos mais precisos de componentes e representações detalhadas de dispositivos de controle. Este avanço consolidou o fluxo de potência como ferramenta estratégica para o planejamento, fornecendo criticidade à tomada de decisão em expansão, investimentos e operação do sistema (KUNDUR, 1994).

No contexto contemporâneo, a análise de fluxo de potência adquire uma importância ainda maior diante dos desafios impostos pela crescente inclusão de fontes renováveis — tais como a solar e a eólica — e pela introdução de controles flexíveis (FACTS). Nestes cenários, a variabilidade temporal das fontes e o perfil dinâmico das cargas exigem ferramentas capazes de responder em tempo real e de lidar com incertezas inerentes à geração e ao consumo (FREITAS & TEIXEIRA, 2023). A emergência dos conceitos de *smart grids* e micro redes redefine a topologia e a operacionalidade das redes, impondo requisitos ainda mais rigorosos às ferramentas de análise.

A possibilidade da comunicação bidirecional, que as redes inteligentes proporcionam entre consumidores, geradores e operadores, aumenta exponencialmente o volume e a veracidade dos dados processados. Consequentemente, o fluxo de potência, que antes se restringia a análises estáticas e pontuais, passa a demandar frequentes atualizações sensíveis a eventos instantâneos, contingências e reconfigurações automáticas. Com o surgimento das micro redes, aumenta o desafio da gestão autônoma de ilhas energéticas, onde decisões descentralizadas dependem de diagnósticos rápidos sobre o estado elétrico local (PEREIRA, 2022).

Diante deste panorama, a busca por métodos mais robustos, escaláveis e adaptativos torna-se imperativa. Técnicas baseadas em inteligência artificial aplicada, especialmente em redes neurais artificiais (RNA), emergem como uma das alternativas mais promissoras para análise de fluxo de potência, sobretudo em situações que demandam processamento em tempo real e adaptação frente às condições dinâmicas e incertas. A aplicação dessas abordagens inovadoras pode redefinir paradigmas tradicionais, contribuindo para a construção de sistemas elétricos mais eficientes e inteligentes.

1.1 Justificativa

A motivação para o desenvolvimento de métodos avançados de análise de fluxo de potência decorre das limitações práticas enfrentadas por operadores e planejadores de sistemas elétricos em cenários reais. As ferramentas tradicionais, embora bem estabelecidas, apresentam desempenho insatisfatório em situações que requerem resposta ágil a eventos contingenciais ou em ambientes altamente dinâmicos, como aqueles caracterizados pela inserção de fontes renováveis e pela reconfiguração constante de topologias (OLIVEIRA et al., 2024).

A lentidão dos cálculos em tempo real é um dos principais problemas práticos de sistemas de grande porte ou submetidos a múltiplas iterações para análise de contingências e otimização operacional. Métodos convencionais como Newton-Raphson e Gauss-Seidel, apesar de robustos, podem demandar até centenas de iterações para atingir a convergência, sobretudo em situações de carregamento crítico ou sob a presença de elementos de controle não lineares (BELPIEDE, 2006). Tal limitação dificulta a implementação de sistemas de supervisão em tempo real, nos quais decisões críticas devem ser tomadas em intervalos de segundos.

Os métodos numéricos tradicionais revelam uma significativa sensibilidade às condições adversas, tais como topologias fortemente malhadas ou mudanças bruscas no perfil de cargas e geração. Podem ocorrer, nesses casos, instabilidades numéricas, falta de convergência ou resultados fisicamente inviáveis, comprometendo assim a confiabilidade dos diagnósticos e das ações corretivas (MARCHESAN et al., 2018). A crescente complexidade agregada pelas tecnologias de Smart grids e microgrids acentua essas limitações, pois introduz múltiplos pontos de injeção e controle descentralizado (SOUZA, 2023).

As RNAs oferecem potencial para mitigar esses desafios ao aprender, a partir de dados históricos e simulados, relações não lineares complexas entre as entradas (parâmetros do sistema) e saídas (resultados do fluxo de potência). Uma vez treinadas, as RNAs são capazes de fornecer estimativas instantâneas do estado do sistema, viabilizando aplicações em tempo real e aumentando a resiliência frente a incertezas e contingências (CRISCIMANI; SANTOS, 2022).

Portanto, a aplicação de RNAs no contexto do fluxo de potência representa uma resposta inovadora e estratégica às demandas dos sistemas elétricos modernos, posicionando-se como alternativa ou complemento aos métodos tradicionais em cenários onde velocidade e adaptabilidade são fatores críticos.

1.2 Objetivos

O objetivo geral é investigar e avaliar a aplicação de redes neurais artificiais na análise de fluxo de potência em sistemas elétricos, comparando seu desempenho e precisão frente aos métodos numéricos tradicionais.

Além disso, os objetivos específicos são os seguintes:

- Revisar e sistematizar a literatura sobre métodos tradicionais de fluxo de potência e sua aplicação em diferentes topologias de redes elétricas. Essa revisão visa consolidar o

entendimento dos algoritmos clássicos, identificando suas limitações e pontos fortes frente a diferentes cenários operacionais.

- Descrever os fundamentos teóricos e arquiteturais das redes neurais artificiais, com foco em arquiteturas do tipo perceptron multicamadas (MLP). A compreensão desses fundamentos permitirá o correto dimensionamento e parametrização das RNAs para a tarefa de previsão de estados elétricos.
- Desenvolver e validar um modelo de RNA para estimativa de variáveis do fluxo de potência, utilizando dados simulados de sistemas elétricos padrão. O modelo será treinado e testado com conjunto de dados representativos visando avaliar sua capacidade preditiva e generalização.
- Comparar o desempenho do modelo de RNA com os métodos de Newton-Raphson e Gauss-Seidel, considerando critérios de velocidade, robustez e precisão. Esta comparação objetiva identificar os contextos nos quais superam ou complementam os métodos tradicionais.
- Analisar a viabilidade de aplicação das RNAs em ambientes de operação em tempo real, discutindo desafios, potencialidades e limitações. Serão explorados aspectos práticos, como escalabilidade, sensibilidade a perturbações e integração com sistemas de supervisão.

2 REVISÃO DE LITERATURA

A revisão bibliográfica estabelece as bases teóricas e técnicas necessárias para o desenvolvimento deste trabalho. São apresentados os conceitos fundamentais de sistemas elétricos de potência, especialmente aqueles relacionados ao fluxo de potência, os principais métodos de solução e às limitações enfrentadas em estudos tradicionais. Em seguida, discute-se o avanço das técnicas de inteligência artificial aplicadas ao setor elétrico, com foco em redes neurais artificiais, aprendizado supervisionado e métodos complementares de análise. Essa fundamentação permite contextualizar a abordagem proposta, bem como justificar o uso de modelos de aprendizado de máquina para estimativa de grandezas elétricas em redes de distribuição e transmissão.

2.1 Fluxo de Potência

A análise de fluxo de potência visa determinar o perfil de tensões, ângulos de fase, potências ativas e reativas em todos os barramentos de um sistema elétrico sob condições operacionais específicas. Dois métodos clássicos predominam nesse contexto: Newton-Raphson e Gauss-Seidel.

2.1.1 Newton-Raphson: Formulação Matricial e Pseudocódigo

O método de Newton-Raphson é largamente utilizado pela sua robustez e rapidez de convergência, especialmente em sistemas de grande porte (KUNDUR, 1994). Fundamenta-se na linearização do sistema em equações não lineares que descrevem o balanço de potência ativa e reativa em cada barra. A potência ativa e a potência reativa injetadas na barra i podem ser escritas, respectivamente, pelas equações (1) e (2):

$$P_i = V_i \sum_{j=1}^n V_j (G_{ij} \cos\theta_{ij} + B_{ij} \sin\theta_{ij}) \quad (1)$$

$$Q_i = V_i \sum_{j=1}^n V_j (G_{ij} \sin\theta_{ij} - B_{ij} \cos\theta_{ij}) \quad (2)$$

onde:

- P_i : Potência ativa injetada na barra i
- Q_i : Potência reativa injetada na barra i
- V_i, V_j : Módulos das tensões nas barras i e j
- $\theta_{ij} = \theta_i - \theta_j$: Diferença angular entre as tensões das barras
- G_{ij} : Parte real da admitância da barra i para a barra j
- B_{ij} : Parte imaginária (susceptância) da admitância entre i e j
- n : Número de barras do sistema.

A solução é obtida via iterações sucessivas, resolvendo-se a cada passo do sistema linear indicado na equação (3).

$$\begin{pmatrix} \Delta P \\ \Delta Q \end{pmatrix} = J \begin{pmatrix} \Delta \theta \\ \Delta V \end{pmatrix} \quad (3)$$

onde J é a matriz jacobiana composta pelas derivadas parciais das potências em relação as tensões e ângulos. O procedimento é repetido até que as correções $\Delta \theta$ e ΔV sejam inferiores a um limiar pré-estabelecido (ϵ), garantindo a convergência.

2.1.2 Gauss-Seidel: Algoritmo Iterativo e Convergência

O método de Gauss-Seidel, apresenta uma convergência mais lenta, especialmente em sistemas malhados ou de grande porte (STOTT & ALSAC, 1974). A estratégia consiste em atualizar iterativamente as tensões barra a barra, utilizando os valores mais recentes disponíveis. A cada atualização da tensão da barra i na iteração $k + 1$ é realizada conforme a equação (4). O critério de parada baseia-se entre as tensões calculadas em iterações sucessivas, usualmente com tolerâncias de ordem 10^{-4} p.u.

$$V_i^{(k+1)} = \frac{1}{Y_{ii}} \left(\frac{P_i - jQ_i}{(V_i^{(k)})^*} - \sum_{j \neq i} Y_{ij} V_j^{(k+1)} \right) \quad (4)$$

2.1.3 Comparação de Desempenho

O Newton-Raphson destaca-se por sua convergência quadrática, menos sensível a complexidade do sistema como número de barras e topologia. Em contraste, Gauss-Seidel apresenta convergência linear resultando em centenas de iterações em grandes sistemas (KUNDUR, 1994). Ao passo que Gauss-Seidel pode não convergir sob condições de carregamentos pesados e sistema fortemente acoplados, Newton-Raphson é mais robusto frente a esses sistemas.

Publicações clássicas, como Stott & Alsac (1974), demonstram que, para sistemas reais, o Newton-Raphson é preferível, apesar da maior complexidade na formação e inversão do jacobiano. Contudo, ambos os métodos partilham limitações inerentes a sua natureza iterativa e ao custo computacional, especialmente em cenários de operação em tempo real.

2.2 Redes Neurais Artificiais

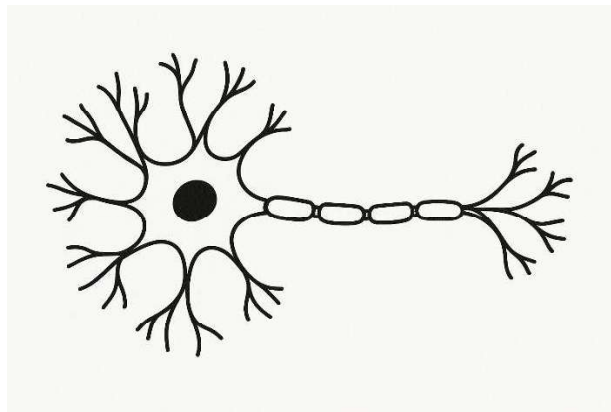
As RNAs são modelos computacionais inspirados no funcionamento do cérebro humano, amplamente aplicados em tarefas de previsão, classificação e estimação de variáveis complexas (HAYKIN, 2009). Dentre as diversas arquiteturas, o perceptron multicamadas (MLP) destaca-se por sua capacidade de aproximar funções não lineares. Elas são compostas

por unidades de processamento interconectadas, chamadas neurônios artificiais, organizadas em camadas.

2.2.1 O Neurônio Artificial: Inspiração Biológica e Modelo Matemático

A unidade fundamental de uma RNA é o neurônio artificial, um modelo matemático simplificado do neurônio biológico. Biologicamente, um neurônio recebe sinais através de seus dendritos, processa esses sinais em seu corpo celular (soma) e, se um limiar de ativação for atingido, dispara um sinal elétrico através do axônio, que é transmitido a outros neurônios através das sinapses. (HAYKIN, 2009). A Figura 1 ilustra essa estrutura funcional básica, utilizada como referência conceitual para o desenvolvimento dos modelos artificiais

Figura 1 – Representação de um neurônio biológico



Fonte: Elaboração Própria (Canva, 2025)

Partes do neurônio:

1. Dendritos – ramificações ao redor do corpo celular (recebem sinais de outros neurônios);
2. Corpo celular (Soma) – parte central arredondada com o núcleo;
3. Núcleo – círculo escuro no centro do corpo celular;
4. Axônio – prolongamento longo que sai do corpo celular;
5. Bainha de mielina – estruturas ovais ao longo do axônio (aumentam a velocidade de condução do impulso);
6. Nódulos de Ranvier – pequenos espaços entre as bainhas de mielina;
7. Terminações axônicas (botões sinápticos) – ramificações finais do axônio.

2.2.2 Funcionamento de um Neurônio Artificial

O neurônio artificial emula o comportamento de um neurônio biológico (HAYKIN, 2009), seguindo as etapas abaixo:

- Entradas (inputs): Recebe Múltiplos sinais de entrada ($x_1, x_2, \dots, x_n$) que podem ser os dados brutos ou as saídas de outros neurônios da camada anterior;
- Pesos Simpáticos: Cada entrada x_i é multiplicada por um peso simpático associado. Esses pesos são os principais parâmetros ajustados durante o aprendizado da rede;

- **Soma Ponderada e Bias:** Os produtos são somados e um termo de bias é frequentemente adicionado a soma ponderada. O bias funciona como um limiar ajustável, permitindo deslocar a função de ativação e aumentando a flexibilidade do modelo. Essa operação está expressa na equação (5):

$$v = u + b = \sum_{i=1}^n x_i w_i + b \quad (5)$$

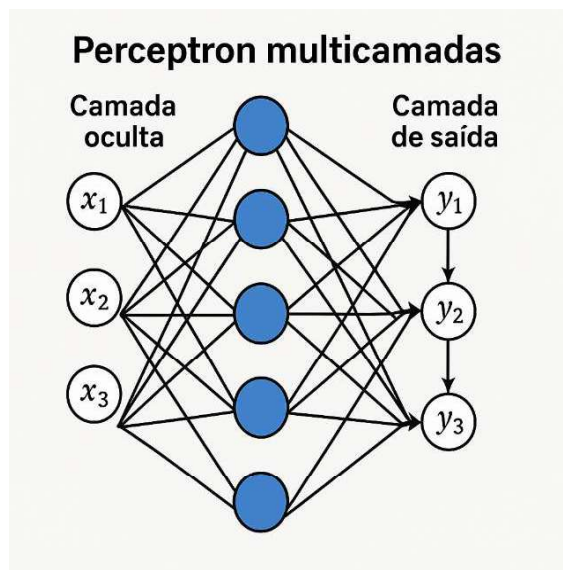
- **Função de Ativação:** O valor é passado por uma função de ativação f que calcula a saída final do neurônio. A função de ativação introduz não linearidade no modelo, o que é crucial para a capacidade da rede de aprender padrões complexos. A saída do neurônio é descrita pela equação (6):

$$y = f(v) = f\left(\sum_{i=1}^n x_i w_i + b\right) \quad (6)$$

2.2.3 Componentes essenciais de uma RNA

Uma RNA típica é organizada em camadas de neurônios interconectados (HAYKIN, 2009). A Figura 2 apresenta a arquitetura geral de um perceptron multicamadas (MLP), modelo clássico utilizado em tarefas de regressão e classificação.

Figura 2 – Perceptron multicamadas



Fonte: Elaboração Própria (Canva, 2025)

- **Camada de Entrada:** é a primeira camada da rede e recebe os dados brutos ou características do problema. Cada neurônio nesta camada geralmente representa uma

variável de entrada, sem realizar cálculos, apenas repassando os valores para a próxima camada;

- **Camada(s) Oculta(s):** São as camadas intermediárias entre a entrada e a saída. É aqui que ocorre a maior parte do processamento e onde a rede aprende representações abstratas dos dados. Uma RNA pode ter uma ou múltiplas camadas ocultas (no caso de redes profundas ou Deep Learning). A complexidade da rede (número de camadas e neurônios por camada) é um hiper parâmetro crucial que afeta sua capacidade de aprendizado e o risco de overfitting;
- **Camada de Saída:** É a última camada da rede e produz o resultado. O número de neurônios na camada de saída depende da tarefa: um neurônio para regressão de valor único, múltiplos neurônios para classificado multiclasse (um por classe).

A Tabela 1 compara algumas das funções de ativação mais comuns:

Tabela 1 – Funções de ativação

Função	Fórmula Matemática	Aplicações Típicas
Linear	$f(x) = x$	Camada de saída (Regressão)
Sigmoide	$f(x) = 1 / (1 + e^{-x})$	Camada de saída (Classificação Binária)
Tanh	$f(x) = \tanh(x) = (e^x - e^{-x}) / (e^x + e^{-x})$	Camadas ocultas
ReLU	$f(x) = \max(0, x)$	Camadas ocultas

Fonte: Elaboração Própria.

2.2.4 Paradigmas de Aprendizado de Máquina para RNAs

Existem três paradigmas principais pelos quais as RNAs podem aprender (HAYKIN, 2009).:

- **Aprendizado Supervisionado:** A rede é treinada com um conjunto de dados onde cada exemplo de entrada possui uma saída desejada correspondente;
- **Aprendizado Não Supervisionado:** A rede é treinada com dados que não possuem rótulos;
- **Aprendizado por Reforço (RL):** Um agente (que pode ser uma RNA) aprende a tomar decisões interagindo com um ambiente.

O objetivo do aprendizado supervisionado é aprender uma função que mapeie as entradas para as saídas corretas, minimizando o erro entre a previsão da rede e o rótulo verdadeiro. Este é o paradigma mais comum para as tarefas de regressão (prever um valor contínuo) e classificação (atribuir uma categoria). A maioria das aplicações de RNAs em fluxo de potência discutidas neste trabalho (previsão de variáveis e estimação de estado) se enquadram neste paradigma.

2.2.5 Treinamento de RNAs: O Algoritmo Backpropagation

O treinamento de uma RNA no paradigma supervisionado visa ajustar os pesos e bias da rede para que ela possa realizar a tarefa desejada com alta precisão. O algoritmo mais

fundamental e amplamente usado para isso é o *backpropagation* (retropropagação de erro) (HAYKIN, 2009).

O objetivo do treinamento é minimizar uma função de perda (*loss*) que quantifica a discrepância entre as saídas previstas pela rede e os valores reais no conjunto de treinamento. O algoritmo *backpropagation* funciona iterativamente da seguinte forma.

1. *Forward Pass*: Um exemplo de entrada de conjuntos de treinamento é apresentado a rede. A entrada propaga-se através das camadas, da entrada para a saída. Cada neurônio calcula sua saída com base nas entradas ponderadas da camada anterior, suas bias e sua função de ativação. Ao final, a camada de saída produz uma previsão;
2. Cálculo do Erro: A previsão da rede é comparada com a saída desejada usando a função de perda pré-definida. O valor da perda indica a quão errada foi a previsão da rede para aquele exemplo;
3. *Backward Pass*: O erro calculado é propagado de volta através da rede, da camada de saída para a camada de entrada. Utilizando a regra da cadeia do cálculo diferencial o algoritmo calcula o gradiente da função de perda em relação a cada peso e bias na rede. O gradiente indica a direção e a magnitude da mudança necessária em cada parâmetro para reduzir o erro. A diferenciabilidade das funções de ativação é crucial nesta etapa para permitir o cálculo dos gradientes;
4. Atualização dos Pesos e Bias: Os pesos e bias são ajustados na direção oposta ao gradiente calculado, geralmente utilizando um algoritmo de otimização. A magnitude do ajuste é controlada por um hiper parâmetro chamado taxa de aprendizado (*learning_rate*).

Esse ciclo é repetido para muitos exemplos do conjunto de treinamento por várias épocas (passagens completas pelo conjunto de treinamento) (HAYKIN, 2009). A cada iteração, a rede gradualmente “aprende” a mapear as entradas para as saídas corretas, ajustando seus parâmetros internos para minimizar a função de perda global.

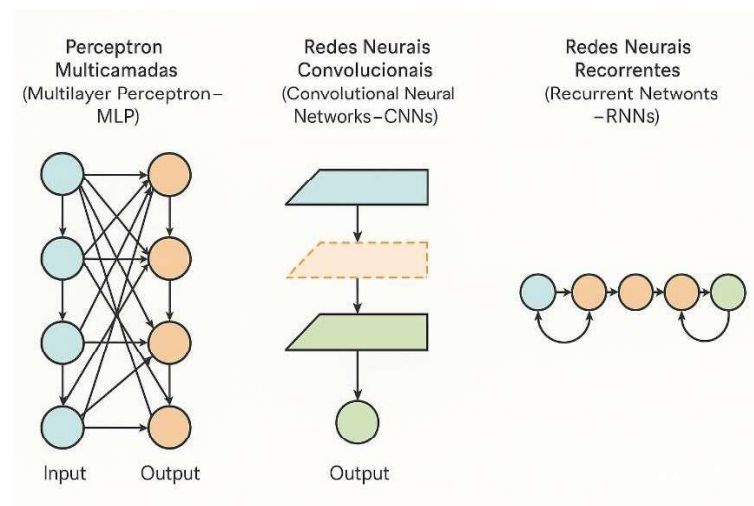
2.2.6 Arquiteturas de RNAs Relevantes para Engenharia

Embora a estrutura básica de camadas interconectadas seja comum, diferentes arquiteturas de RNAs foram desenvolvidas para lidar eficientemente com tipos específicos de dados e problemas. Algumas das mais relevantes para aplicação em engenharia, incluído sistemas elétricos são:

- Perceptron Multicamadas (MLP): É a arquitetura de RNA *feedforward* mais clássica, consistindo em uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída, onde cada neurônio em uma camada está conectado a todos os neurônios da camada seguinte. MLPs são capazes de aprender funções não lineares complexas e são adequadas para dados estruturados e problemas gerais de classificação. Em sistemas elétricos de potência (SEPs) são frequentemente usadas como baseline ou para tarefas de previsão direta baseadas em dados vetoriais (GOODFELLOW; BENGIO; COURVILLE, 2016).
- Redes Neurais Convolucionais (CNNs): São arquiteturas especializadas para processar dados com uma topologia de grade conhecida, como imagens (2D) ou séries temporais/sequenciais (1D). CNNs utilizam camadas especiais:

- Camadas Convolucionais: Aplicam filtros que deslizam sobre a entrada para detectar padrões locais, como bordas em imagens ou padrões em sequencias;
- Camadas de Agrupamento: Reduzem a dimensionalidade espacial das representações, tornando o modelo mais robusto a pequenas variações na posição das features (invariância e translações) e diminuindo a carga computacional;
- Camadas totalmente conectadas: Geralmente usadas no final da rede para realizar a classificação ou regressão com base nas características extraídas. Em SEPs, CNNs são aplicadas em análise de imagens (inspeção visual, termografia), processamento de dados de PMUs (Phasor Measurement Units) formatados como “imagens” de estado do sistema, e análise de contingência ou estresse do sistema, onde padrões espaço-temporais são relevantes.

Figura 3 – Arquiteturas de RNAs



Fonte: Elaboração Própria (Canva, 2025)

A escolha da arquitetura da RNA não deve ser aleatória. Ela é fundamentalmente guiada pela estrutura e natureza dos dados inerentes ao problema específico do SEP que deseja resolver, conforme ilustrado na Figura 3. MLPs são adequadas para dados tabulares ou vetoriais onde não há uma estrutura espacial ou temporal explícita forte. CNNs exploram a localidade espacial ou sequencial e a invariância a translações, sendo ideais para imagens ou dados que podem ser representados em grades. RNNs são projetadas para capturar dependências temporais em sequência. Arquiteturas ainda mais especializadas como as Graph Neural Networks (GNNs) são adequadas para dados que representam explicitamente a topologia relacional da rede elétrica. Essa adequação entre a arquitetura e a estrutura dos dados é um passo crítico na modelagem e reflete uma crescente sofisticação na aplicação de RNAs em engenharia de sistemas de potência, movendo-se de abordagens genéricas para soluções mais adaptadas ao domínio (HAYKIN, 2009).

2.3 Redes Neurais Aplicadas em Modelagem de Sistemas Elétricos

A aplicação de redes neurais artificiais (RNAs) em sistemas elétricos tem se mostrado uma das estratégias mais eficazes para lidar com problemas de alta não linearidade e complexidade computacional. Essas técnicas vêm sendo usadas para previsão de carga, estimação de estado, modelagem de fluxo de potência e otimização operacional (ZHANG et al., 2021; SINGH et al., 2021).

A seguir, são descritas as principais abordagens relevantes para este trabalho.

2.3.1 Estimação de estado e aprendizado supervisionado

A estimação de estado é um dos campos onde as RNAs mais evoluíram. Trabalhos recentes demonstram que modelos *feedforward* são capazes de mapear diretamente grandezas medidas (potência ativa e reativa, tensões e ângulos) em estados estimados, sem necessidade de resolver o sistema de equações iterativamente (SINGH et al., 2021).

Além de reduzirem o tempo de execução, as redes apresentam boa capacidade de generalização e robustez frente a ruídos de medição. Essa característica é especialmente útil para sistemas com medidores limitados ou SCADA incompletos.

2.3.2 Modelagem do fluxo de potência

O problema de fluxo de potência é amplamente utilizado como referência para testar a capacidade de aprendizado e generalização das RNAs. Zhang et al. (2021) demonstraram que um modelo MLP treinado com resultados do método de Newton-Raphson reproduz as tensões e ângulos nodais com erro médio quadrático inferior a 1%, reduzindo o tempo de simulação em até uma ordem de grandeza.

De modo semelhante, Li et al. (2023) utilizaram redes compactas otimizadas para representar as equações AC de forma eficiente, comparando seus resultados com métodos lineares DC e obtendo maior fidelidade. Esses avanços são fundamentais para aplicações em tempo real, como despacho econômico e monitoramento dinâmico de micro redes, e inspiram diretamente a metodologia adotada neste trabalho.

2.3.3 Hibridização com algoritmos clássicos e aprendizado de características

Abordagens híbridas integram RNAs e algoritmos de aprendizado de máquina clássicos para aprimorar convergência e interpretabilidade. CHEN et al. (2022) propuseram o uso de *Random Forest* para gerar estimativas iniciais de tensão e ângulo no método de Newton-Raphson, acelerando a convergência e reduzindo falhas numéricas. Outros autores, como OLIVEIRA et al. (2024), aplicaram *Extreme Gradient Boosting* (XGBoost) para identificar a importância de variáveis de entrada em sistemas de potência, o que se relaciona diretamente com a engenharia de *features* utilizada neste trabalho — onde a seleção e combinação de variáveis derivadas (carga, geração e hora) aumentam a precisão das RNAs.

2.3.4 Aplicações emergentes e perspectivas

Recentemente, arquiteturas mais avançadas como CNNs, LSTMs e Physics-Informed Neural Networks (PINNs) vêm sendo aplicadas à previsão de carga e ao controle de redes inteligentes (SOUZA et al., 2020). Esses modelos incorporam restrições físicas e temporais

diretamente na estrutura da rede, garantindo coerência com as leis elétricas e maior interpretabilidade. A tendência é que, nos próximos anos, as RNAs sejam integradas a sistemas operacionais de potência em larga escala, tanto para previsão e controle como para diagnóstico automático e análise de estabilidade (LI et al., 2023).

Essa evolução demonstra que a integração entre aprendizado supervisionado e modelos físicos está consolidando uma nova geração de sistemas elétricos inteligentes — base conceitual da proposta desenvolvida neste TCC.

2.4 Bibliotecas em Python

Um *pipeline* de aprendizado supervisionado é uma estrutura sequencial que organiza todas as etapas necessárias para transformar dados brutos em previsões úteis. Seu objetivo é garantir reprodutibilidade, padronização e eficiência em todo o processo de modelagem. Em vez de executar cada etapa isoladamente, o pipeline integra o fluxo completo de operações em uma cadeia lógica e automatizada, permitindo que diferentes modelos sejam avaliados sob as mesmas condições. O pipeline foi inteiramente desenvolvido em Python 3.11, com bibliotecas de código aberto que compõem um ecossistema consolidado para simulação elétrica, aprendizado de máquina e análise de dados. A seguir, são detalhadas as principais ferramentas empregadas e o papel de cada uma no projeto.

2.4.1 Pandapower — Modelagem e Simulação Elétrica

O Pandapower é a base da modelagem elétrica deste projeto. Ele fornece uma interface de alto nível para criação, análise e otimização de redes de energia elétrica. (THOMAS et al., 2018).

- Função: executa os cálculos de fluxo de potência AC e DC, curtos-circuitos e estudos de contingência.
- Aplicação no projeto: o sistema IEEE 14 barras foi modelado e simulado com o método Newton-Raphson para gerar as tensões e ângulos de referência usados como base de dados no treinamento das redes neurais.

2.4.2 Pandas — Estruturação e Manipulação de Dados

O Pandas foi utilizado para leitura, transformação e análise dos dados provenientes da curva de carga da ONS e dos cenários simulados.

- Função: oferece estruturas de dados como DataFrame e ferramentas de data wrangling eficientes. (MCKINNEY, 2010)
- Aplicação no projeto: integração entre a curva horária de carga e as potências nodais, filtragem de outliers e junção com atributos temporais (hora, dia e mês).

2.4.3 Processamento Numérico Vetorial

As operações vetoriais foram executadas com o NumPy. (HARRIS et al., 2020).

- Função: manipulação eficiente de vetores, matrizes e operações matriciais necessárias ao pré-processamento e treinamento.
- Aplicação: normalização de entradas e saídas, cálculo de métricas de erro e suporte ao backend das bibliotecas de aprendizado profundo.

2.4.4 Aprendizado Supervisionado e Pré-Processamento

O Scikit-learn é uma das bibliotecas mais amplamente utilizadas para aprendizado de máquina em Python. (PEDREGOSA et al., 2011)

- Função: oferece modelos supervisionados (como Random Forest e XGBoost), ferramentas de validação cruzada, padronização e avaliação de desempenho;
- Aplicação: Random Forest: usada para medir a importância das variáveis e orientar a seleção de features mais influentes;
- StandardScaler: empregado para normalização dos dados de entrada e saída;
- Validação cruzada K-Fold: aplicada para avaliar a robustez dos modelos.

2.4.5 TensorFlow e Keras — Modelagem Neural Profunda

O TensorFlow, com interface de alto nível Keras, foi a base para construção, treinamento e validação das Redes Neurais Artificiais (RNA-MLP). (ABADI et al., 2016)

- Função: fornece infraestrutura para aprendizado profundo, permitindo treinamento acelerado em CPU/GPU;
- Aplicação:
- Construção de duas redes MLP independentes para prever magnitudes e ângulos de tensão;
- Emprego do otimizador Adam, função de perda Huber, e early stopping para evitar sobre ajuste. exportação de modelos em formato .h5 para uso em inferências rápidas.

2.4.6 XGBoost — Ensemble Gradient Boosting

O XGBoost complementa a análise das importâncias de atributos e serve como modelo comparativo de alto desempenho (CHEN; GUESTRIN, 2016).

- Função: implementa o algoritmo de Gradient Boosted Trees com regularização L1/L2, oferecendo interpretabilidade;
- Aplicação: comparação com Random Forest e RNA, reforçando a análise estatística das variáveis críticas para as tensões nodais.

2.4.7 Matplotlib e Seaborn — Visualização e Análise Gráfica

As bibliotecas Matplotlib e Seaborn foram usadas para gerar visualizações analíticas e comparativas.

Os gráficos estatísticos foram produzidos com o Seaborn (WASKOM, 2021). As visualizações foram elaboradas com o Matplotlib (HUNTER, 2007).

- Função: construção de gráficos de desempenho (MAE, RMSE, R^2), visualizações 3D dos hiperparâmetros e correlação entre variáveis;
- Aplicação: criação de painéis visuais que ilustram o aprendizado do modelo e o comportamento das tensões nas barras da rede.

O propósito é demonstrar que a união entre métodos simbólicos (físicos) e empíricos (data-driven) pode gerar um sistema de inferência rápido, preciso e confiável para aplicações em tempo real.

3 MÉTODO PROPOSTO PARA ANÁLISE DE FLUXO DE POTÊNCIA USANDO REDES NEURAI

O presente trabalho propõe um pipeline híbrido de aprendizado supervisionado para a análise de fluxo de potência em sistemas elétricos, integrando modelagem física clássica (via Pandapower) e aprendizado de máquina (Redes Neurais Artificiais).

O método foi concebida para investigar como encontrar uma arquitetura neural eficiente e interpretável, e não apenas qual rede neural melhor se ajusta ao problema.

3.1 Curva de Carga

Nesta seção descreve-se o processo de obtenção, filtragem e análise da curva de carga horária da região Sudeste do Sistema Interligado Nacional (SIN) referente ao ano de 2023, com posterior geração de gráficos que evidenciam padrões de comportamento da demanda elétrica ao longo do tempo

3.1.1 Obtenção dos dados

A curva de carga horária foi obtida diretamente do portal de Dados Abertos do Operador Nacional do Sistema Elétrica (ONS). O arquivo referente ao ano de 2023 foi baixado em formato CSV, contendo registros horários da carga (em MW) para os quatro subsistemas: Norte, Nordeste, Sul e Sudeste.

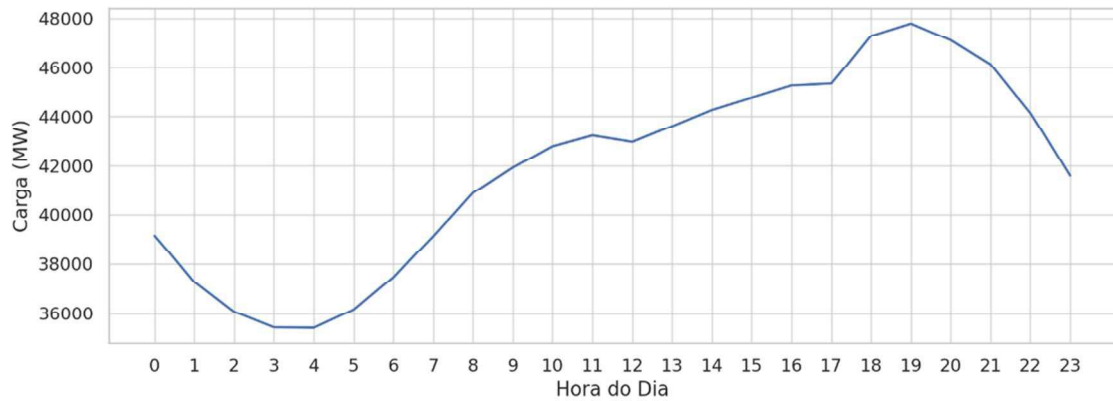
3.1.2 Filtragem Para a Região Sudeste

Com a base carregada em Python, realizamos a filtragem das entradas referentes à região Sudeste. Em seguida, foram criadas colunas como hora (0 a 23h), dia da semana (0 = segunda-feira, 6 = domingo) e mês (1 a 12), essas colunas servem como base para análises estatísticas e visualização da variabilidade temporal da carga.

3.1.3 Geração dos Gráficos Estatísticos

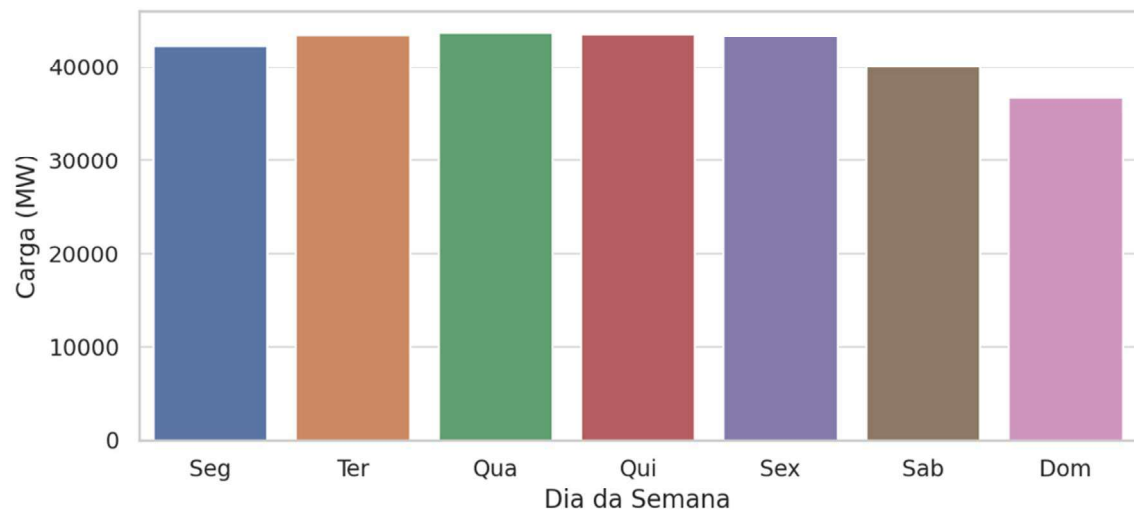
Foram produzidos três gráficos centrais para a caracterização do comportamento do consumo elétrico na região Sudeste. A Figura 4 apresenta a média horária da carga ao longo do dia, permitindo observar padrões típicos de demanda. A Figura 5 mostra a média por dia da semana, destacando diferenças entre dias úteis e finais de semana. A Figura 6 exibe a média de carga mês a mês, evidenciando variações sazonais.

Figura 4 – Média de carga por hora do dia



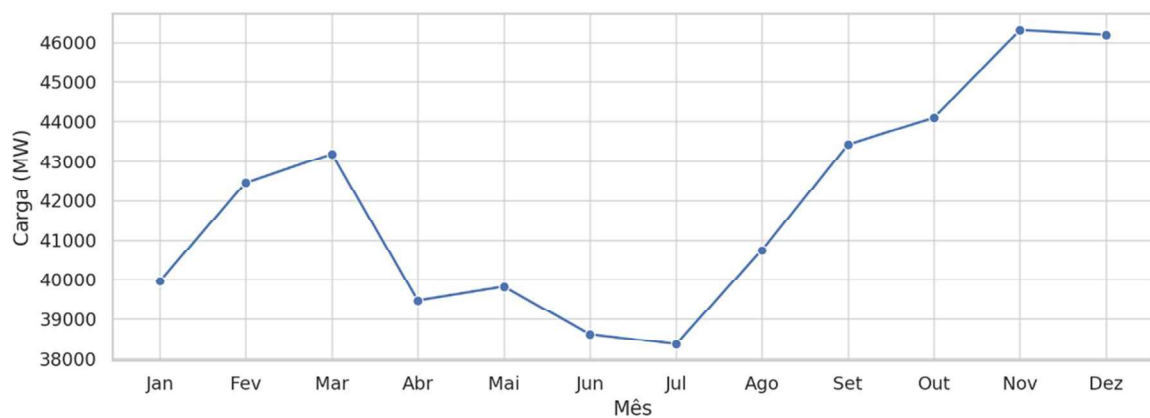
Fonte: Elaboração própria a partir de dados da ONS (Google Colab, 2025).

Figura 5 – Média de carga por dia da semana



Fonte: Elaboração própria a partir de dados da ONS (Google Colab, 2025).

Figura 6 – Média de carga mês-a-mês



Fonte: Elaboração própria a partir de dados da ONS (Google Colab, 2025).

3.1.4 Utilização dos Dados

Essa curva de carga filtrada e estruturada servirá de base para multiplicar diretamente as cargas ativas e reativas do sistema IEEE 14 de barras, gerando cenários realistas para treinamento e validação da rede neural. Essa abordagem busca uma ponte entre dados reais do setor elétrico brasileiro e modelos computacionais inteligentes, alinhando teoria e prática de forma aplicada.

3.2 Modelagem Elétrica

3.2.1 Geração de Cenários Físicos

O sistema de teste IEEE de 14 barras foi adotado como base.

Os valores de carga ativa e reativa foram escalonados conforme a Curva de Carga Horária da Região Sudeste (ONS, 2023), composta por 8760 registros horários.

Dessa forma, para cada instante temporal t , obtiveram-se novos valores de potência ativa e reativa segundo as Equações (7) e (8):

$$P_i(t) = P_i(0) \times f(t) \quad (7)$$

$$Q_i(t) = Q_i(0) \times f(t) \quad (8)$$

Onde $f(t)$ é o fator de carga normalizado em relação à demanda máxima anual.

Cada cenário foi processado via solver AC Newton-Raphson no Pandapower, resultando em um conjunto de dados contendo:

- Entradas: P_i , Q_i , hora, dia da semana e mês;
- Saídas: V_i , θ_i (magnitudes e ângulos de tensão).

Para modelagem do fluxo de potência cada cenário gerado foi resolvido com o solver AC Newton–Raphson do pacote Pandapower. Esse processo gerou um conjunto de dados com 8760 amostras físicas. As entradas do conjunto são: potências ativas e reativas (P_i , Q_i), hora do dia, dia da semana e mês; as saídas são as magnitudes de tensão (V_i) e os ângulos de tensão (θ_i).

3.2.2 Pré-processamento e Normalização

Os dados brutos foram tratados para:

- Eliminação de inconsistências;
- Padronização numérica com StandardScaler;
- Inclusão de variáveis temporais (hora, dia e mês) como features normalizadas em $[0,1]$.

As bases resultantes foram separadas em:

- X : entradas do modelo
- $y_{vm} \rightarrow 9$ tensões em p.u.
- $y_{va} \rightarrow 14$ ângulos de fase (graus)

A divisão dos dados seguiu a proporção:

- 70% treino, 15% validação e 15% teste.

No contexto de aprendizado supervisionado, o conjunto de dados é dividido em três partes com funções distintas. O conjunto de treino é utilizado para ajustar os parâmetros internos do modelo, permitindo que ele aprenda os padrões presentes nas amostras. O conjunto de validação é empregado durante o processo de ajuste de hiperparâmetros, funcionando como uma etapa intermediária para avaliar o desempenho do modelo enquanto ele ainda está em desenvolvimento; sua função é evitar sobre ajuste ao fornecer um critério externo para comparar diferentes arquiteturas e configurações. Por fim, o conjunto de teste é utilizado apenas após o término completo do treinamento, servindo como medida final e imparcial de desempenho. Essa separação garante que a avaliação do modelo reflita sua capacidade real de generalização, evitando que os resultados sejam influenciados pelo conhecimento prévio das amostras usadas durante o processo de aprendizado.

3.3 Modelos de Aprendizado Supervisionado

Após a geração dos cenários físicos, o método recorre a modelos de aprendizado supervisionado para aproximar o fluxo de potência. Além da previsão, serviu para calcular importâncias de atributos, identificando quais variáveis exercem maior influência sobre as tensões nodais.

3.3.1 Random Forest

O Random Forest foi usado como modelo basal interpretável. Além de prever as saídas, ele fornece a importância dos atributos, evidenciando quais variáveis (potências, hora, dia ou mês) têm maior influência sobre as tensões nodais. Esse diagnóstico conecta as grandezas físicas com a inferência estatística.

3.3.2 XGBoost

O modelo Extreme Gradient Boosting foi empregado para confirmar e complementar os resultados do Random Forest. A técnica incorpora regularização ℓ_1/ℓ_2 e shrinkage e apresenta métricas de gain e cover que reforçam a consistência das importâncias calculadas. Ele também serve como uma baseline competitiva em termos de desempenho.

3.3.3 Modelagem Neural (RNA – MLP)

Arquitetura: a rede neural utilizada é um Perceptron Multicamadas (MLP) com duas camadas ocultas contendo oito neurônios cada, função de ativação ReLU e camada de saída linear. O treinamento foi feito com o otimizador Adam e função de perda Huber. Treinaram-se duas redes independentes:

- RNA_vm $\rightarrow$ estima magnitudes de tensão;
- RNA_va $\rightarrow$ estima ângulos de tensão.

A etapa neural representa o núcleo de aprendizado profundo do pipeline, refinado a partir das importâncias detectadas nas árvores.

3.3.4 Validação

Para garantir uma avaliação robusta da capacidade de generalização da RNA, utilizou-se validação cruzada do tipo K-Fold com $K = 5$. Nesse procedimento, o conjunto completo de amostras é dividido em cinco subconjuntos de tamanho semelhante. Em cada iteração, quatro subconjuntos são utilizados para treinar o modelo, enquanto o subconjunto restante é usado exclusivamente para teste. O processo é repetido cinco vezes, alternando o subconjunto destinado ao teste até que todas as amostras tenham sido avaliadas exatamente uma vez. Ao final, as métricas de desempenho são calculadas para cada *fold* e, posteriormente, agregadas (por média e variância), permitindo observar tanto o desempenho médio quanto a estabilidade do modelo em diferentes particionamentos. Essa abordagem impede que o resultado dependa de uma única divisão dos dados e fornece um diagnóstico mais confiável sobre eventuais problemas de sobreajuste ou sensibilidade às amostras. As métricas de desempenho utilizadas foram o Erro Absoluto Médio (MAE), a Raiz do Erro Quadrático Médio (RMSE) e o coeficiente de determinação R^2 , definidos pelas equações (9), (10) e (11), respectivamente.

$$MAE = \frac{1}{n} \sum |y_i - \hat{y}_i| \quad (9)$$

$$RMSE = \sqrt{\frac{1}{n} \sum (y_i - \hat{y}_i)^2} \quad (10)$$

$$R^2 = 1 - \frac{\sum (y_i - \hat{y}_i)^2}{\sum (y_i - \bar{y})^2} \quad (11)$$

Essas métricas asseguraram confiabilidade estatística e replicabilidade do aprendizado em múltiplos subconjuntos de dados.

Avaliação dos resultados: os resultados foram analisados em três níveis:

- Curvas Previsto $\times$ Real para verificar a correlação direta;
- Erro médio por barra, fornecendo um mapa espacial das discrepâncias;
- Tabela comparativa que reúne os desempenhos de Random Forest, XGBoost e rede neural.

3.4 Grid Search

Esta subseção descreve o procedimento de busca em grade (grid search) adotado para seleção de hiperparâmetros dos modelos de redes neurais utilizados na predição das grandezas elétricas do sistema IEEE de 14 barras. A estratégia foi aplicada separadamente para dois problemas supervisionados:

- (i) Predição dos módulos de tensão por barra (*bus_vm_pu*);
- (ii) Predição dos ângulos de tensão por barra (*bus_va_degree*).

3.4.1 Espaço de Busca

A busca em grade contemplou combinações dos seguintes hiperparâmetros para modelos MLP totalmente conectados:

- Função de ativação: {linear, ReLU, Tanh}.
- Número de camadas ocultas: {1, 2, 3}.
- Neurônios por camada: {8, 16, 32, 64, 128}.
- *Batch size*: 32.
- Otimizador: Adam (parâmetros padrão do Keras).
- Função de perda (treino): MAE (erro absoluto médio).
- *Early stopping*: monitor em *val_loss* com *patience* = 15 e restauração dos melhores pesos.

4 RESULTADOS E DISCUSSÃO

Neste capítulo são apresentados os resultados obtidos com o pipeline híbrido proposto e discutidas as implicações metodológicas e físicas do uso de redes neurais artificiais em análise de fluxo de potência.

A metodologia proposta integra um solver clássico (Newton–Raphson) com um modelo de aprendizado supervisionado, treinado a partir de 8.760 cenários horários da curva de carga da região Sudeste de 2023. Cada cenário contém as potências ativa (P) e reativa (Q) consumidas nas barras do sistema IEEE de 14 barras, além das variáveis temporais *hora* e *dia da semana*. As saídas desejadas são as magnitudes e ângulos de tensão em cada barra. O objetivo é avaliar se a RNA consegue reproduzir o solver físico com alta precisão e em tempo reduzido, proporcionando uma alternativa para aplicações em tempo real.

4.1 Grid Search

Esta subseção apresenta os resultados do grid search para os dois modelos independentes (Vm e VA), seguido da análise crítica dos desempenhos, do comportamento por barra e de implicações para o uso prático no fluxo de potência.

4.1.1 Sumário dos desempenhos

A Tabela 2 consolida os Top-20 modelos para cada saída, ordenados por MAE em escala real, com desempate por RMSE e R^2 .

Tabela 2 – Resultados Grid Search Vm

Ativação	Neurônios	Camadas	Val MAE	MAE (p.u.)	RMSE (p.u.)	R^2
ReLU	8	3	0,000434	0,000429	0,000536	0,9486
Tanh	8	3	0,000477	0,000471	0,000583	0,9481
ReLU	16	3	0,000609	0,000606	0,000757	0,9001
ReLU	8	2	0,000756	0,000725	0,000925	0,8583
ReLU	64	2	0,000765	0,000762	0,000988	0,7792
Tanh	16	3	0,000838	0,000829	0,001043	0,8460
Tanh	128	3	0,000843	0,000837	0,001026	0,8575
Tanh	8	2	0,000868	0,000841	0,001175	0,7195
linear	128	3	0,000937	0,000920	0,001266	0,6764
linear	8	1	0,000948	0,000926	0,001176	0,7880
ReLU	32	3	0,000928	0,000933	0,001179	0,7708
ReLU	128	2	0,000936	0,000934	0,001169	0,7811
Tanh	64	3	0,001027	0,001012	0,001238	0,6918
ReLU	32	2	0,001063	0,001059	0,001391	0,7125
ReLU	16	2	0,001095	0,001081	0,001423	0,7326

Tanh	128	2	0,001115	0,001096	0,001402	0,7035
linear	64	3	0,001124	0,001117	0,001371	0,7244
Tanh	32	3	0,001132	0,001122	0,001478	0,4917
ReLU	128	3	0,001158	0,001141	0,001458	0,6920
ReLU	8	1	0,001166	0,001170	0,001546	0,6947

Fonte: Elaboração Própria.

Tabela 3 – Resultados Grid Search VA

Ativação	Neurônios	Camadas	Val_MAE	MAE (°)	RMSE (°)	R ²
Tanh	8	3	0,004493	0,004057	0,016294	0,9284
ReLU	8	3	0,004449	0,004188	0,005881	0,9286
ReLU	32	1	0,004416	0,004238	0,005512	0,9286
Tanh	16	3	0,004674	0,004459	0,011477	0,9285
Tanh	8	2	0,005843	0,004686	0,017164	0,9284
Tanh	32	2	0,005052	0,004775	0,007781	0,9285
Tanh	8	1	0,005629	0,005024	0,012020	0,9285
Tanh	16	1	0,006134	0,005239	0,012310	0,9285
linear	32	1	0,005773	0,005545	0,008175	0,9285
linear	16	1	0,005957	0,005650	0,008367	0,9285
ReLU	16	1	0,006016	0,005696	0,008921	0,9285
ReLU	8	1	0,006286	0,005729	0,008189	0,9285
Tanh	32	1	0,006779	0,005801	0,013343	0,9285
linear	16	2	0,006140	0,005949	0,008460	0,9285
ReLU	8	2	0,006507	0,006381	0,009600	0,9285
linear	8	1	0,006676	0,006517	0,010274	0,9285
Tanh	32	3	0,007139	0,006733	0,014408	0,9285
Tanh	16	2	0,008047	0,006813	0,020718	0,9283
Tanh	64	3	0,006982	0,006813	0,009325	0,9285
ReLU	64	1	0,007138	0,006913	0,008994	0,9285

Fonte: Elaboração Própria.

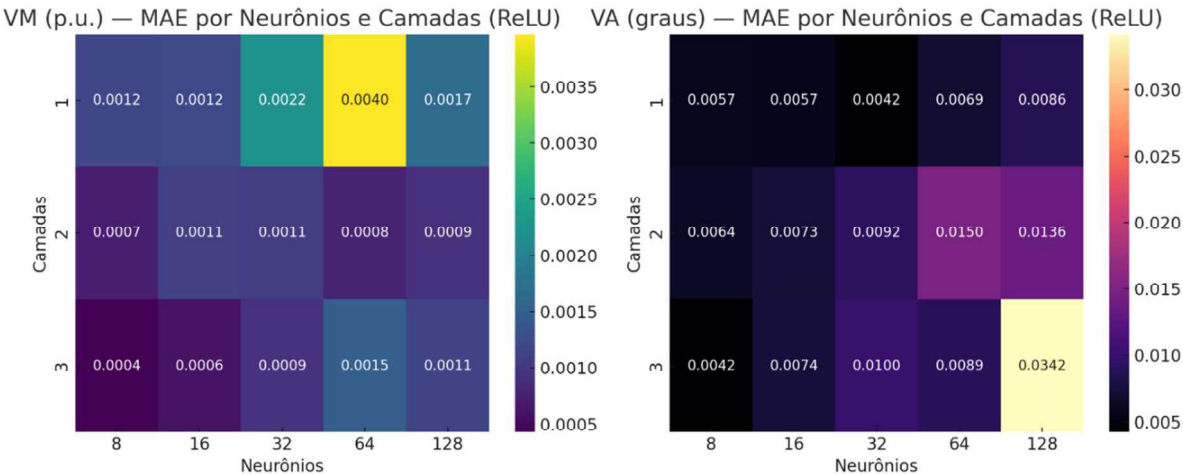
Interpretação geral

- Modelos Vm (módulo de tensão): em geral, arquiteturas com 2–3 camadas e 8–16 neurônios por camada mostraram bom compromisso viés-variância, com MAE baixo e

R^2 elevado. A ativação ReLU tendeu a apresentar ligeira vantagem em Vm por facilitar a modelagem de relações monotônicas com cargas;

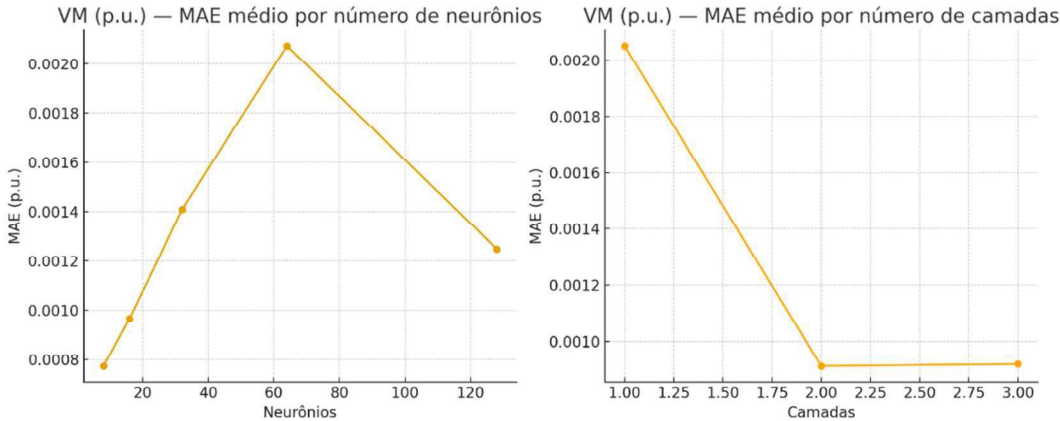
- Modelos VA (ângulo de tensão): o alvo apresenta maior sensibilidade a variações operativas e topológicas, refletindo em RMSE relativamente superior a Vm. Ativações Tanh ou ReLU produziram resultados competitivos; quando há maior dispersão, o ajuste de neurônios tende a ser mais relevante que o número de camadas, consistindo com práticas de ajuste fino em MLP.

Figura 7 – Heatmap de MAE para as arquiteturas



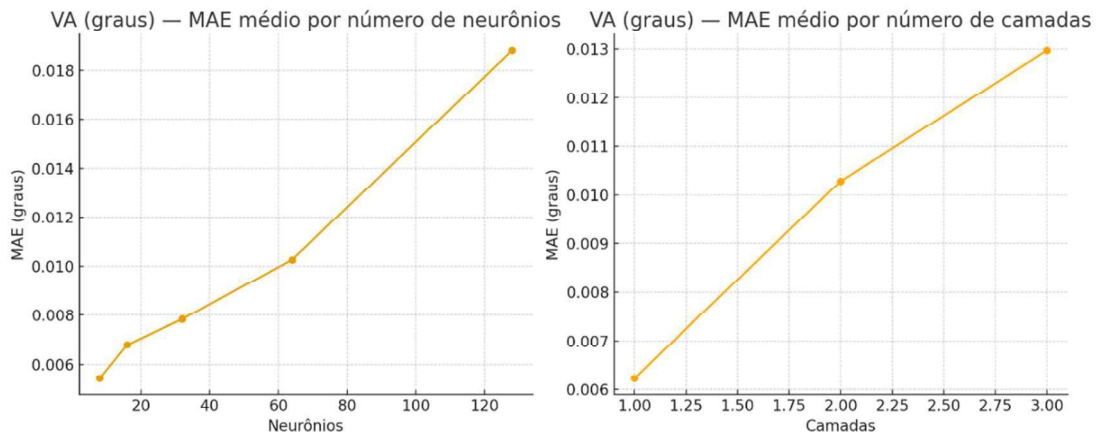
Fonte: Elaboração própria (Google Colab, 2025).

Figura 8 – MAE médio de Vm (p.u.)



Fonte: Elaboração própria (Google Colab, 2025).

Figura 9 – MAE médio de VA (graus)



Fonte: Elaboração própria (Google Colab, 2025).

4.1.2 Implicações e recomendações

A função de ativação ReLU foi analisada em profundidade por meio de um *grid search* envolvendo diferentes números de neurônios e camadas ocultas. O objetivo foi identificar as topologias que apresentaram o melhor desempenho na predição das variáveis V_m (magnitude da tensão em p.u.) e V_a (ângulo da tensão em graus).

A Figura 7 apresenta mapas de calor (*Heatmap*) relacionando o MAE médio obtido para cada combinação de neurônios e camadas ocultas. No caso da variável V_m , observou-se que arquiteturas com 8 a 16 neurônios e duas a três camadas alcançaram os menores erros médios, com valores de MAE da ordem de 10^{-4} p.u., indicando excelente capacidade de generalização. Para V_a , as topologias ótimas concentraram-se igualmente em redes compactas (8–32 neurônios), com erro médio absoluto inferior a 0,005 graus, evidenciando estabilidade e ausência de sobre ajuste relevante.

A análise visual dos Heatmap demonstra que, à medida que a complexidade da rede aumenta (maior número de neurônios ou camadas), o ganho de desempenho torna-se marginal ou até decrescente, sugerindo que o problema em estudo é bem condicionado e de baixa dimensionalidade efetiva após a etapa de engenharia de *features*.

As Figuras 8 e 9 apresentam os gráficos de tendência do erro médio em função do número de neurônios e de camadas ocultas para cada variável.

No caso de V_m , o erro médio cresce levemente conforme o aumento do número de camadas, reforçando que modelos rasos apresentam melhor equilíbrio entre viés e variância. Para V_a , o comportamento é semelhante, com uma faixa de estabilidade entre 8 e 32 neurônios, acima da qual há saturação do desempenho e incremento de erro, possivelmente associado ao aumento da variância do modelo.

Essa tendência indica que estruturas simples são suficientes para representar as relações não lineares do fluxo de potência, especialmente considerando que as entradas já foram previamente normalizadas e refinadas por meio da engenharia de *features*.

Em resumo, a função de ativação ReLU apresentou comportamento consistente e robusto tanto para V_m quanto para V_a , com erros médios baixos e alta estabilidade entre repetições. As melhores topologias encontradas foram:

- V_m : 8 neurônios, 3 camadas ocultas, $MAE \approx 0,000434$ p.u.
- V_a : 8 neurônios, 3 camadas ocultas, $MAE \approx 0,004057$ graus.

Esses resultados demonstram que o modelo de rede neural é capaz de reproduzir o comportamento físico do sistema elétrico com excelente precisão, validando a adequação da abordagem de aprendizado supervisionado na modelagem do fluxo de potência.

4.2 Desempenho da RNA

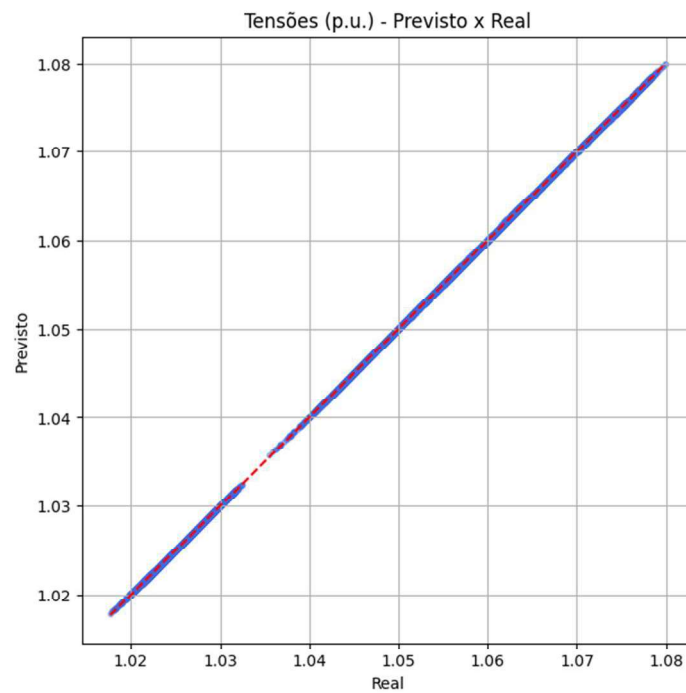
O tempo de inferência da RNA foi de 1,31 s para todos os cenários, enquanto o solver Newton–Raphson levou 471,79 s. Deve-se levar em conta que esse é o tempo de inferência da RNA comparado ao tempo de montagem, iteração e resolução necessárias para resolver o fluxo de potência com o método de Newton-Raphson.

As Figuras 10 e 11 ilustram a nuvem de dispersão entre os valores obtidos via Newton-Raphson e os estimados pela RNA para as magnitudes de tensão em p.u e ângulos em graus. Os pontos alinham-se em torno da bissetriz, denotando a capacidade do modelo de reproduzir o solver físico. Para os ângulos, observa-se maior espalhamento, sobretudo em barras sujeitas a grandes variações de carga, refletindo a sensibilidade desses estados à distribuição reativa do sistema (KUNDUR, 1994).

4.2.1 Avaliação

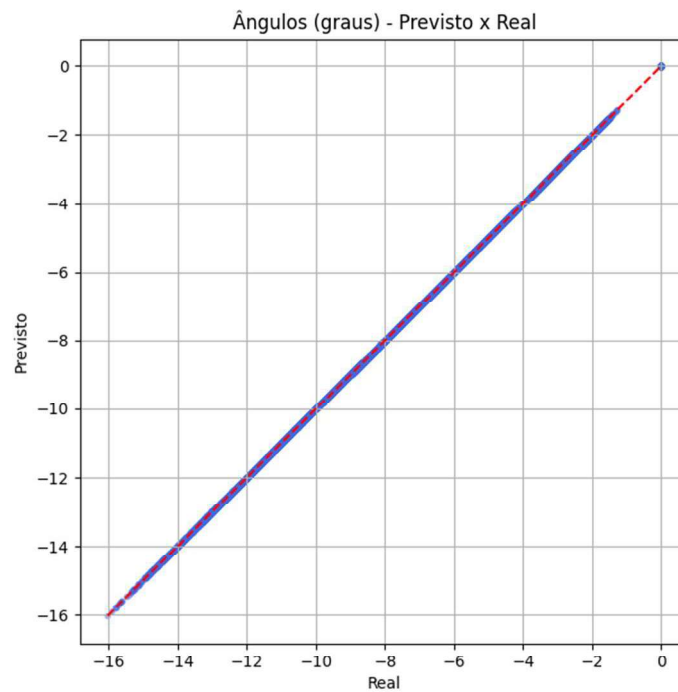
- Dispersão real $\times$ previsto: as Figuras 10 e 11 exibem nuvens próximas da diagonal para V_m e para V_a ;
- Erro por barra: barras com maior variabilidade de carga/fluxo ou proximidade do *slack* tendem a concentrar maior erro em V_a ; para V_m , a distribuição de erro é mais homogênea. Isso sugere que futuros aprimoramentos podem incluir *features* topológicas (ex.: grau da barra, centralidade) e/ou representação angular $\sin./\cos$.
- Erro médio por barra: As Figuras 12 e 13 mostram o MAE médio para cada barra do sistema. Para as magnitudes de tensão, os erros são distribuídos de forma relativamente homogênea, refletindo a suavidade típica do comportamento de V_m . Para os ângulos, observa-se maior erro em barras adjacentes ao slack ou sujeitas a variações mais pronunciadas de carga e fluxo. Esse comportamento sugere que futuros aprimoramentos podem incluir *features* topológicas — como grau da barra, centralidade ou impedâncias relativas — e até mesmo representações trigonométricas ($\sin/\cos$) para os ângulos, reduzindo a descontinuidade associada ao domínio angular.

Figura 10 – Previsto x Real de Vm (p.u.)



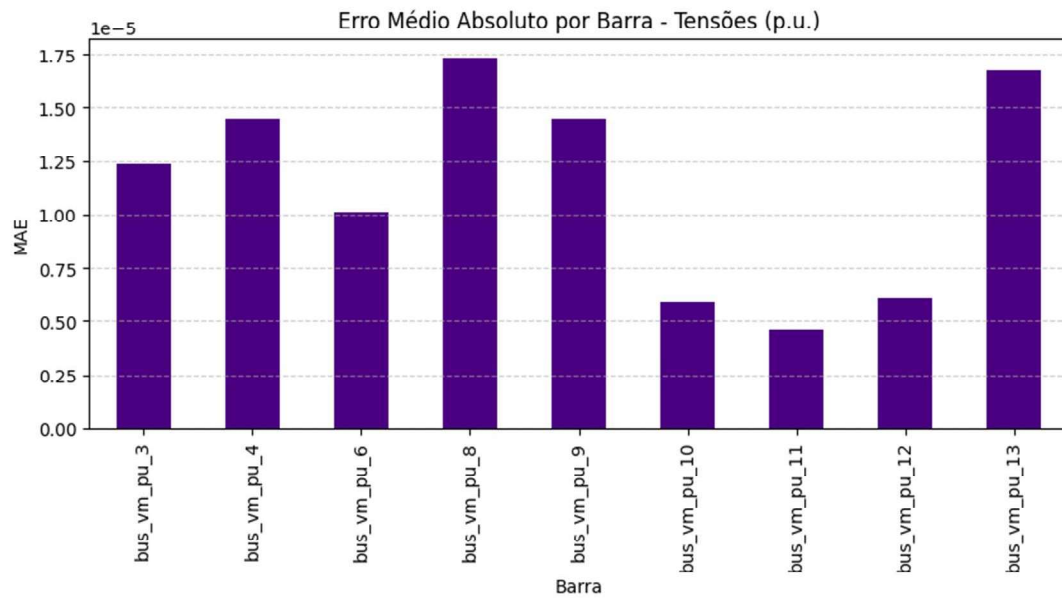
Fonte: Elaboração própria (Google Colab, 2025).

Figura 11 – Previsto x Real de VA (graus)



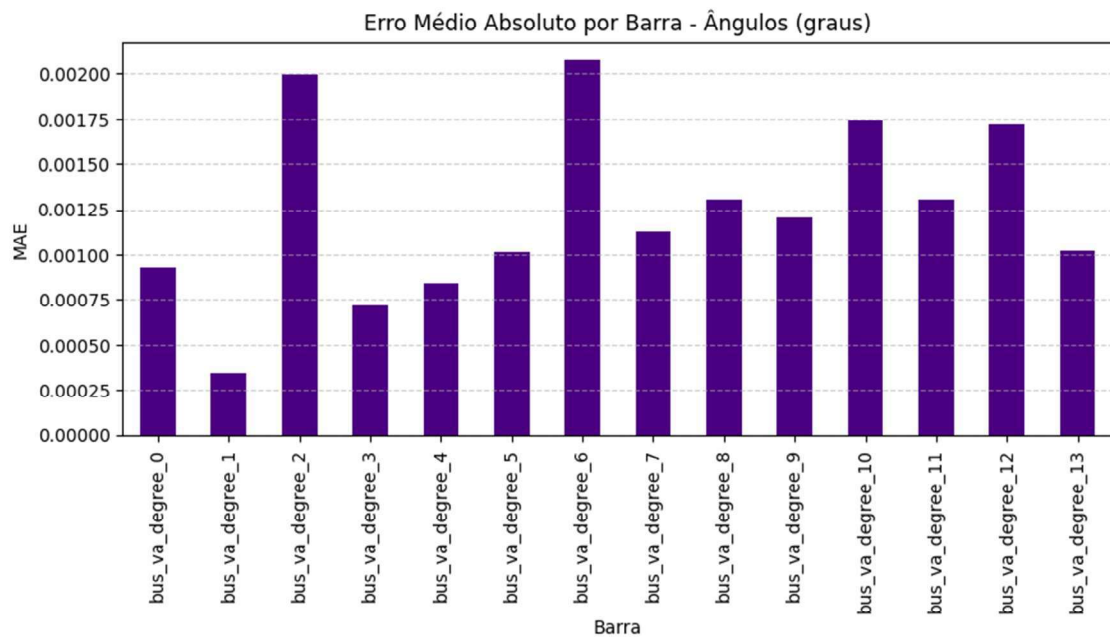
Fonte: Elaboração própria (Google Colab, 2025).

Figura 12 – MAE médio por barra (Vm)



Fonte: Elaboração própria (Google Colab, 2025).

Figura 13 – MAE médio por barra (VA)



Fonte: Elaboração própria (Google Colab, 2025).

Resultado da RNA: ReLU 3 camadas 8 neurônios:

- Desempenho Tensões (Vm):
- MAE: 0,003794 | RMSE: 0,006110 | R²: 0,999963
- Desempenho Ângulos (VA):
- MAE: 0,001472 | RMSE: 0,002108 | R²: 0,928567

4.3 Importância das Variáveis

Para entender a contribuição de cada feature para o erro residual da RNA, foram treinados modelos de Random Forest e XGBoost sobre o erro médio absoluto das previsões. Esses resultados nos permite investigar quais variáveis são críticas para a precisão da RNA. A identificação dessas viáveis-chave indicadas na Tabela 4 sugere que medições precisas nessas barras e a inclusão de atributos temporais são essenciais para melhorar modelos de fluxo de potência.

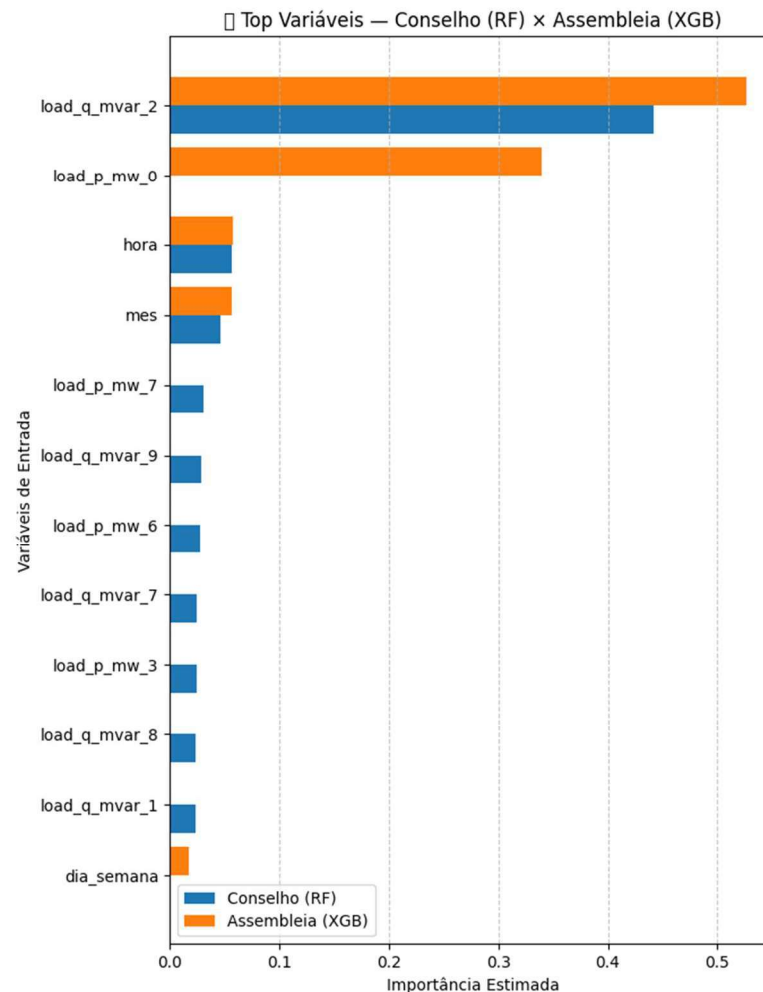
Tabela 4 – Importância das Features

Nº	Variável	Importância RF	Importância XGB
1	load_q_mvar_1	0,023225	0,000000
2	load_q_mvar_8	0,023891	0,000000
3	load_p_mw_3	0,024730	0,000000
4	load_q_mvar_7	0,024984	0,000000
5	load_p_mw_6	0,028352	0,000000
6	load_q_mvar_9	0,028675	0,000000
7	load_p_mw_7	0,030662	0,000000
8	mes	0,046499	0,057206
9	hora	0,056576	0,058076
10	dia_semana	0,000000	0,017993
11	load_p_mw_0	0,000000	0,339829
12	load_q_mvar_2	0,441678	0,526896

Fonte: Elaboração Própria.

A regressão do erro médio da RNA por meio de algoritmos de árvores (Random Forest) e gradiente (XGBoost) forneceu percepções sobre a contribuição de cada entrada para o desvio residual. Ambas as técnicas convergiram para a potência reativa da barra 2 como a principal responsável pelos erros: o Random Forest atribuiu-lhe 44,17 % de importância e o XGBoost, 52,26 %. Esta barra faz parte de um dos nós de carga mais sensíveis, e pequenas oscilações em Q provocam alterações perceptíveis nos ângulos de tensão de várias barras, explicando a dificuldade do modelo em capturar completamente tal relação. O XGBoost também evidenciou a potência ativa da barra 0 e as variáveis temporais “hora” e “dia da semana”, sugerindo que o comportamento dinâmico da curva de carga influencia o padrão residual. A análise de importância reforça a visão de que, apesar de a RNA aproximar a função fluxograma, certos estados são particularmente sensíveis a mudanças de reativos, recomendando estratégias de medição mais densas nesses pontos. A Figura 14 ilustra o gráfico de importancia estimada das variáveis com maior importancia.

Figura 14 – Gráfico de importância de variáveis



Fonte: Elaboração própria (Google Colab, 2025).

4.4 Robustez e Generalização

A avaliação da robustez e da capacidade de generalização da RNA foi conduzida por meio de validação cruzada K-Fold, metodologia recomendada em cenários onde a distribuição das amostras pode influenciar o desempenho do modelo (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

Na abordagem K-Fold, o conjunto de cenários é dividido em K subconjuntos mutuamente exclusivos. Em cada iteração, um subconjunto é usado como teste, enquanto os demais compõem o treino, garantindo que todas as amostras sejam avaliadas exatamente uma vez. Neste trabalho utilizou-se $K = 5$, valor amplamente aceito em problemas de regressão com quantidade moderada de dados (GOODFELLOW; BENGIO; COURVILLE, 2016).

Os resultados mostraram baixa variabilidade entre as dobras, indicando que a RNA mantém desempenho consistente independentemente da partição. A Tabela 5 contém as métricas dos folds.

Tabela 5 – Métricas K-Fold

Fold	MAE	RMSE	R ²
1	0,001880	0,004410	0,9286
2	0,001672	0,002097	0,9286
3	0,002617	0,005240	0,9286
4	0,001994	0,002608	0,9286
5	0,002841	0,003631	0,9286

Fonte: Elaboração Própria.

A métrica de R² apresentou valores estáveis de 0,9286 para VA, confirmando que o modelo generaliza bem para diferentes subconjuntos de operação. Essa estabilidade estatística evidencia que a RNA não depende fortemente de uma configuração específica de treinamento e que sua capacidade preditiva é intrinsecamente robusta à variação natural dos dados.

Do ponto de vista físico-elétrico, esse comportamento é coerente com a estrutura do problema: em regiões normais de operação, o fluxo de potência apresenta suavidade e regularidade nas relações entre carga e estado, favorecendo modelos supervisionados baseados em aproximações contínuas (MARCHESAN; CUNHA; SILVA, 2018). Assim, a validação cruzada confirma que o modelo captura de forma confiável os padrões dominantes sem apresentar sobreajuste significativo.

4.5 Discussão

Os experimentos conduzidos demonstram que o pipeline híbrido proposto, combinando engenharia de features, grid search extensivo e análise física dos estados elétricos permite que modelos de rede neural artificial reproduzam o fluxo de potência com elevada precisão e significativa redução de tempo computacional.

O grid search realizado para Vm e Va analisou mais de 60 combinações entre número de neurônios, profundidade e funções de ativação. Para Vm, as melhores arquiteturas concentraram-se em redes compactas com 2 a 3 camadas e 8 a 16 neurônios por camada, alcançando erros médios absolutos na ordem de 10⁻⁴ p.u. e coeficientes de determinação elevados. O mesmo padrão ocorreu para Va, cujo melhor desempenho foi obtido por uma rede de 8 neurônios e 3 camadas, com MAE de aproximadamente 0,004 graus. Os mapas de calor evidenciaram que aumentos de complexidade não produzem ganhos significativos, reforçando que o problema apresenta baixa dimensionalidade efetiva após a etapa de pré-processamento.

A análise gráfica entre valores previstos e reais confirmou boa aderência para Vm e comportamento ligeiramente mais disperso para Va, coerente com a maior sensibilidade angular às variações de reativos. Os erros por barra mostraram distribuição homogênea para Vm e maior concentração em pontos próximos ao slack ou a barras com variação reativa acentuada para Va. Esses resultados sugerem que futuras melhorias podem incluir atributos topológicos ou representações trigonométricas para capturar a estrutura angular de maneira mais suave.

O desempenho temporal ilustra a principal vantagem operacional da abordagem: a RNA inferiu todos os cenários em 1,31 s, enquanto o método clássico Newton–Raphson exigiu 471,79 s, diferença expressiva no contexto de aplicações em tempo real ou análises de grande escala.

A análise de importância de variáveis revelou padrões consistentes entre Random Forest e XGBoost. A potência reativa da barra 2 foi identificada como o principal determinante do erro residual, apontando para uma região do sistema onde pequenas oscilações influenciam múltiplos ângulos. Variáveis temporais como hora e dia da semana também apresentaram importância relevante, indicando que a dinâmica da curva de carga exerce papel significativo no comportamento residual mesmo após o ajuste da RNA.

Por fim, a validação cruzada K-Fold confirmou a estabilidade do modelo, com baixa variabilidade entre as dobras e R^2 constante em 0,9286 para V_a . Esse comportamento demonstra que a RNA não depende de uma divisão específica dos dados e generaliza de forma consistente para diferentes subconjuntos operativos, o que reforça sua robustez para aplicação prática.

Em conjunto, os resultados evidenciam que o modelo neural é capaz de replicar o solver físico com boa precisão, estabilidade estatística e ganhos substanciais de desempenho computacional, oferecendo um caminho viável para estimativas rápidas de estados e aplicações operacionais em redes de potência.

5. CONCLUSÃO

Este trabalho investigou a aplicação de redes neurais artificiais (RNAs) na análise de fluxo de potência, propondo um pipeline híbrido que integra modelagem física clássica e aprendizado supervisionado. A abordagem adotada buscou não apenas comparar métodos numéricos tradicionais com modelos de aprendizado de máquina, mas também compreender de maneira sistemática como a engenharia de features, a seleção de variáveis e a escolha arquitetural influenciam o desempenho e a estabilidade das soluções. A utilização do sistema IEEE de 14 barras, aliado a uma base realista de demanda elétrica derivada da curva de carga da região Sudeste, permitiu construir um ambiente experimental robusto e representativo das condições operativas de sistemas elétricos reais.

Os resultados obtidos demonstram que a RNA é capaz de reproduzir, com precisão adequada para aplicações operacionais, as magnitudes e ângulos de tensão do fluxo de potência obtidos pelo método Newton–Raphson. Mais do que isso, o estudo evidenciou que a combinação entre modelagem física e aprendizado estatístico reduz substancialmente a necessidade de arquiteturas profundas ou hiper parametrizações complexas, indicando que a estrutura do próprio problema confere simplicidade ao espaço de soluções. A expressiva redução no tempo de inferência, comparada aos métodos clássicos, revela um potencial significativo para aplicações que exigem alta velocidade, como estimativa de estado em tempo real, detecção de anomalias e análises de grande escala.

Além de confirmar a viabilidade técnica do uso de RNAs no fluxo de potência, o trabalho contribui ao apresentar uma metodologia reprodutível, transparente e fundamentada tanto em princípios elétricos quanto em práticas modernas de aprendizado de máquina. O pipeline desenvolvido integra seleção de atributos, validação cruzada, análise interpretável e avaliação comparativa, compondo um framework completo que pode servir de base para estudos futuros. Conclui-se que modelos neurais, quando construídos com rigor e alinhados aos fenômenos físicos subjacentes, representam uma alternativa eficaz e complementar aos métodos tradicionais, com potencial para ampliar a eficiência, a adaptabilidade e a inteligência das redes elétricas modernas.

A partir dos resultados encontrados, diversos caminhos naturais de evolução se apresentam para ampliar o alcance e a precisão da abordagem. Uma das direções mais promissoras envolve aprofundar a representação da topologia do sistema elétrico. Embora as variáveis de carga e o contexto temporal tenham capturado grande parte da dinâmica, a inclusão de atributos estruturais — como distâncias elétricas, impedâncias relativas e medidas de centralidade — pode refinar o desempenho em regiões particularmente sensíveis, sobretudo nas barras que apresentaram maior erro angular.

Outra possibilidade reside na forma de representar os ângulos. O uso direto de valores em graus impõe descontinuidades que podem ser suavizadas por uma codificação trigonométrica baseada em seno e cosseno. Essa transformação tende a favorecer a aprendizagem de relações contínuas e pode reduzir a variância observada em pontos críticos. O método também pode ser estendido a redes mais complexas e com múltiplos níveis de tensão.

A integração entre estimadores clássicos e modelos neurais constitui outro desdobramento relevante. Enquanto RNAs oferecem rapidez e boa aproximação, estimadores tradicionais garantem segurança numérica e coerência física rigorosa. A combinação dessas abordagens pode gerar soluções híbridas mais robustas, especialmente em ambientes de operação com incertezas variáveis nas medições ou com reconfigurações frequentes da rede.

Por fim, a viabilidade do método em aplicações de tempo real pode ser explorada mediante ensaios com janelas móveis de atualização, execução em hardware otimizado e experimentos em cenários com eventos abruptos. Essa etapa permitirá avaliar o desempenho do modelo além do ambiente experimental e representará um avanço significativo para a adoção prática de estimadores baseados em aprendizado de máquina em sistemas elétricos de potência.

6. Referências

- ABADI, M. et al. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. 2016. Disponível em: <https://www.tensorflow.org/>. Acesso em: 9 nov. 2025.
- BASTOS, D. Como funciona um neurônio. *Revista Mundo Estranho*, São Paulo: Editora Abril, 2014. Disponível em: <https://super.abril.com.br/mundo-estranho/como-funciona-um-neuronio/>. Acesso em: 9 nov. 2025.
- BELPIEDE, F. Estudo comparativo entre métodos de fluxo de potência aplicados a sistemas elétricos de potência. 2006. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Federal de Itajubá, 2006.
- CHEN, T.; GUESTRIN, C. XGBoost: A Scalable Tree Boosting System. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. San Francisco: ACM, 2016. p. 785–794. DOI: 10.1145/2939672.2939785.
- CRISCIMANI, R.; SANTOS, L. A. Aplicação de redes neurais artificiais para estimação de estado em sistemas elétricos de potência. *Revista Brasileira de Engenharia Elétrica*, v. 29, n. 2, p. 45–59, 2022. DOI: 10.21575/rbee.v29i2.2649.
- FREITAS, R.; TEIXEIRA, P. Fluxo de potência em ambientes com fontes renováveis e FACTS: uma abordagem atualizada. *Revista de Engenharia Elétrica e Computação*, v. 18, n. 1, p. 25–38, 2023.
- GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep Learning. Cambridge, MA: MIT Press, 2016. Disponível em: <https://www.deeplearningbook.org/>. Acesso em: 9 nov. 2025.
- HARRIS, C. R. et al. Array programming with NumPy. *Nature*, v. 585, p. 357–362, 2020. DOI: 10.1038/s41586-020-2649-2.
- HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. The Elements of Statistical Learning: Data Mining, Inference, and Prediction. 2. ed. New York: Springer, 2009. DOI: 10.1007/978-0-387-84858-7.
- HAYKIN, S. Neural Networks and Learning Machines. 3. ed. New York: Prentice Hall, 2009.
- HUNTER, J. D. Matplotlib: A 2D Graphics Environment. *Computing in Science & Engineering*, v. 9, n. 3, p. 90–95, 2007. DOI: 10.1109/MCSE.2007.55.
- KUNDUR, P. Power System Stability and Control. New York: McGraw-Hill, 1994.
- LI, Y.; WU, L.; HUANG, Z. Physics-informed neural networks for AC power flow: A hybrid data–physics approach for fast and accurate estimation. *Electric Power Systems Research*, v. 219, p. 109187, 2023. DOI: 10.1016/j.epsr.2023.109187.
- MARCHESAN, A.; CUNHA, M.; SILVA, E. Análise de convergência de métodos iterativos para fluxo de potência em redes malhadas. *Revista Controle & Automação*, v. 29, n. 4, p. 486–495, 2018.

MCKINNEY, W. Data Structures for Statistical Computing in Python. In: *Proceedings of the 9th Python in Science Conference (SciPy 2010)*. Austin, TX, 2010. p. 56–61. DOI: 10.25080/Majora-92bf1922-00a.

OLIVEIRA, R. M. et al. Aplicação de técnicas híbridas para análise de fluxo de potência em sistemas elétricos inteligentes. *IEEE Latin America Transactions*, v. 22, n. 4, p. 551–560, 2024.

ONS – Operador Nacional do Sistema Elétrico. Curva de Carga Horária 2023. Dados Abertos. Disponível em: <https://dados.ons.org.br/dataset/curva-carga>. Acesso em: 9 nov. 2025.

PEDREGOSA, F. et al. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

PEREIRA, L. S. Controle e gerenciamento de microgrids no contexto de redes inteligentes. *Revista Engenharia Elétrica*, v. 15, n. 2, p. 77–93, 2022.

SINGH, R.; SHARMA, A.; PAL, B. C. State estimation in power systems using artificial neural networks: A fast non-iterative approach. *Electric Power Components and Systems*, v. 49, n. 15–16, p. 1458–1470, 2021. DOI: 10.1080/15325008.2021.1958024.

SOUZA, D. M. Desafios da convergência em sistemas de potência com inserção de geração distribuída. *Revista de Energia e Sustentabilidade*, v. 9, n. 1, p. 13–25, 2023.

STOTT, B.; ALSAC, O. Fast Decoupled Load Flow. *IEEE Transactions on Power Apparatus and Systems*, v. PAS-93, n. 3, p. 859–869, 1974. DOI: 10.1109/TPAS.1974.293985.

THOMAS, S. et al. Pandapower – an Open-Source Python Tool for Convenient Modeling, Analysis, and Optimization of Electric Power Systems. *IEEE Transactions on Power Systems*, v. 33, n. 6, p. 6510–6521, 2018. DOI: 10.1109/TPWRS.2018.2829021.

WASKOM, M. L. Seaborn: Statistical Data Visualization. *Journal of Open Source Software*, v. 6, n. 60, p. 3021, 2021. DOI: 10.21105/joss.03021.

ZHANG, H.; ZHAO, J.; SUN, H.; GUO, Q. Power flow calculation using deep neural networks trained with Newton–Raphson solutions. *IEEE Transactions on Power Systems*, v. 36, n. 5, p. 4753–4763, 2021. DOI: 10.1109/TPWRS.2021.3071739.