UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

VICTOR CESAR SIMÕES DE OLIVEIRA

PREVISÃO DE DEMANDA DE ENERGIA ELÉTRICA COM A UTILIZAÇÃO DE REDES NEURAIS ARTIFICIAIS

PONTA GROSSA

2025

VICTOR CESAR SIMÕES DE OLIVEIRA

PREVISÃO DE DEMANDA DE ENERGIA ELÉTRICA COM A UTILIZAÇÃO DE REDES NEURAIS ARTIFICIAIS

Forecasting of electric energy demand utilizing Artificial Neural Networks

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

Orientador:
Prof. Dr. Hugo Valadares Siqueira

PONTA GROSSA

2025

VICTOR CESAR SIMÕES DE OLIVEIRA

PREVISÃO DE DEMANDA DE ENERGIA ELÉTRICA COM A UTILIZAÇÃO DE REDES NEURAIS ARTIFICIAIS

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

Data de aprovação: 21 de fevereiro de 2025

Prof. Dr. Hugo Valadares Siqueira
Universidade Tecnológica Federal do Paraná

Profa. Dra. Fernanda Cristina Corrêa
Universidade Tecnológica Federal do Paraná

Profa. Dra. Marcella Scoczynski Ribeiro Martins
Universidade Tecnológica Federal do Paraná

PONTA GROSSA

2025

AGRADECIMENTOS

Primeiramente, agradeço a Deus pelas oportunidades que me foram apresentadas.

Agradeço também à minha família por todo amor, apoio e educação que me deram, e à Sabrina, meu amor, por ter acreditado em mim.

Ao Prof. Hugo, meu orientador e amigo, pelas reuniões, correções e conhecimento transmitido.

Ao LICON, laboratório onde nasceu este trabalho e funcionou como minha base dentro da universidade.

Aos amigos que fiz dentro do curso de Engenharia Elétrica pelo companheirismo e por tornarem a jornada mais leve.

Aos amigos do Residencial Vivace que fizeram com que eu nunca ficasse realmente sozinho em uma cidade estranha para mim.

E ao Campeão dos Campeões, Corinthians, por toda a alegria proporcionada!

RESUMO

A importância da energia elétrica na vida das pessoas é inegável, pois ela contribui para o desenvolvimento e bem-estar de uma sociedade. Por isso, é necessário buscar métodos e ferramentas a fim de se ter um planejamento para o fornecimento seguro de energia elétrica para a população. Assim sendo, neste trabalho foi proposto a utilização de Redes Neurais Artificiais (RNA) para tal. Foram desenvolvidas RNAs do tipo Perceptron de Múltiplas Camadas (MLP), Funções de Base Radial (RBF), Máquina de Aprendizado Extremo (ELM), Máquina de Estado de Eco (ESN) e Memória de Longo-Curto Prazo (LSTM) para realizar a previsão de séries temporais de demanda de energia elétrica para os estados do Paraná, de Santa Catarina e do Rio Grande do Sul. Os resultados obtidos mostram a viabilidade da proposta, pois todas as redes obtiveram bons resultados, apresentando erros aceitáveis e gerando curvas de previsão condizentes com a realidade, sendo a MLP a rede com a melhor previsibilidade.

Palavras-chave: Demanda de energia elétrica; redes neurais artificiais; séries temporais.

ABSTRACT

The importance of electric power in people's lives is undeniable, as it contributes to the development and well-being of a society. Therefore, it is necessary to seek methods and tools to ensure safe planning for the supply of electric power to the population. Thus, this work proposes the use of Artificial Neural Networks (ANN) for this purpose. Multiple Layer Perceptron (MLP), Radial Basis Function (RBF), Extreme Learning Machine (ELM), Echo State Network (ESN), and Long-Short Term Memory (LSTM) neural networks were developed to forecast time series of electric power demand for the states of Paraná, Santa Catarina, and Rio Grande do Sul. The results obtained show the feasibility of the proposal, as all the networks achieved good results, presenting acceptable errors and generating prediction curves consistent with reality, with the MLP being the network with the best predictability.

Keywords: electric power demand; artificial neural networks; time series.

LISTA DE FIGURAS

Figura 1 – Representação de um neurônio biológico 14
Figura 2 – Representação de um neurônio artificial 15
Figura 3 – Estrutura de rede *feedforward* 19
Figura 4 – RNA desdobrada em três épocas pela técnica de BPTT. 20
Figura 5 – Arquitetura da rede MLP 23
Figura 6 – Arquitetura da rede RBF 25
Figura 7 – Arquitetura da rede ELM 28
Figura 8 – Arquitetura da rede ESN 29
Figura 9 – Arquitetura da rede LSTM 32

LISTA DE GRÁFICOS

Gráfico 1 – *Dataset* Paraná 34
Gráfico 2 – *Dataset* Santa Catarina 35
Gráfico 3 – *Dataset* Rio Grande do Sul 35
Gráfico 4 – Boxplot dos MSEs para a série do PR 41
Gráfico 5 – Previsão da ESN - PR 41
Gráfico 6 – Boxplot dos MSEs para a série do SC 42
Gráfico 7 – Previsão da ELM - SC 42
Gráfico 8 – Boxplot dos MSEs para a série do RS 43
Gráfico 9 – Previsão da ESN - RS 44

LISTA DE TABELAS

Tabela 1 – Estatística descritiva das bases de dados 34
Tabela 2 – Funções de ativação definidas 36
Tabela 3 – Hiperparâmetros finais - Paraná 39
Tabela 4 – Hiperparâmetros finais - Santa Catarina 39
Tabela 5 – Hiperparâmetros finais - Rio Grande do Sul 39
Tabela 6 – Erros - Paraná 40
Tabela 7 – Erros - Santa Catarina 41
Tabela 8 – Erros - Rio Grande do Sul 43

LISTA DE ABREVIATURAS E SIGLAS

ELM	*Extreme Learning Machine*
EPE	Empresa de Pesquisa Energética
ESN	*Echo State Network*
LSTM	*Long-Short Term Memory*
MAE	*Mean Absolute Error*
MAPE	*Mean Absolute Percentage Error*
MLP	*Multilayer Perceptron*
MSE	*Mean Squared Error*
RBF	*Radial Basis Function*
ReLU	*Rectified Linear Unit*
RMSE	*Root Mean Squared Error*
RNA	Rede Neural Artificial
Wh	Watt-hora

SUMÁRIO

1 INTRODUÇÃO 11
1.1 Objetivos 11

2 FUNDAMENTAÇÃO TEÓRICA 12
2.1 Demanda de energia elétrica 12
2.2 Séries temporais 12
2.3 Inspiração biológica 13
2.4 Neurônio artificial 14
2.4.0.1 Função de ativação 17
2.5 Redes Neurais Artificiais 18
2.5.1 Características das redes neurais 18
2.5.1.1 Redes recorrentes e não recorrentes 19
2.5.1.2 Mecanismos de treinamento 20
2.5.1.3 Validação 22
2.5.2 Perceptron de Múltiplas Camadas (MLP) 22
2.5.3 Função de Base Radial (RBF) 24
2.5.4 Máquinas de Aprendizado Extremo (ELM) 27
2.5.5 Rede com Estado de Eco (ESN) 28
2.5.6 Memória de Longo-Curto Prazo (LSTM) 31

3 MATERIAL E MÉTODOS 34
3.1 Base de dados 34
3.2 Pré-processamento 35
3.3 Ajustes dos hiperparâmetros 36
3.4 Métricas 37

4 RESULTADOS E DISCUSSÃO 39
4.1 Hiperparâmetros finais 39
4.2 Avaliação das previsões 40
4.2.1 Paraná 40
4.2.2 Santa Catarina 40
4.2.3 Rio Grande do Sul 42
4.3 Considerações finais 43

5 CONCLUSÕES 45

REFERÊNCIAS 46

1 INTRODUÇÃO

A energia elétrica contribui para o desenvolvimento da sociedade e o bem-estar das pessoas. A matriz de energia brasileira é composta em grande parte por energia hidrelétrica, o que faz com que exista uma insegurança no seu fornecimento em períodos de baixa precipitação pluviométrica, por exemplo, como já ocorreu no passado.

Dessa maneira, fica clara a necessidade de um planejamento para o fornecimento de energia elétrica. Para isso, é preciso buscar procedimentos e métodos de análise para compreender padrões e tendências no seu consumo.

A previsão de demanda de energia elétrica é uma ferramenta que faz esse papel. É possível utilizar Redes Neurais Artificiais (RNAs) para fazer a previsão de uma série temporal de dados com consumos de energia registrados.

Neste sentido, este estudo irá abordar a aplicação de diferentes Redes Neurais Artificiais para a realização da previsão de demanda de energia elétrica nos estados do Paraná, de Santa Catarina e do Rio Grande do Sul. As redes utilizadas neste trabalho são o Perceptron de Múltiplas Camadas (MLP), Função de Base Radial (RBF), Máquina de Aprendizado Extremo (ELM), Rede com Estados de Eco (ESN) e Memória de Longo-Curto Prazo (LSTM).

1.1 Objetivos

O presente trabalho tem como objetivo realizar um estudo sobre a previsão de demanda de energia elétrica nos estados do Paraná, de Santa Catarina e do Rio Grande do Sul, utilizando diferentes topologias de Redes Neurais Artificiais.

Este trabalho também será composto pelos seguintes objetivos específicos:

- Implementar e avaliar diferentes topologias de RNA;
- Fazer uma análise comparativa entre elas;
- Justificar a escolha da melhor topologia.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Demanda de energia elétrica

A energia elétrica contribui para o desenvolvimento de todas as regiões do Brasil e para a qualidade de vida de toda a população, pois a eletricidade é uma necessidade básica e quem não tem acesso a ela é colocado à margem da inclusão social (Tidre; Biase; Silva, 2013).

A universalização do acesso à energia elétrica ajuda a reduzir desigualdades sociais e regionais, proporcionando oportunidades iguais para todos. Em áreas rurais e remotas, a eletrificação pode transformar comunidades, oferecendo novas possibilidades de desenvolvimento e integração (Valois Coelho; Cartaxo, 2004).

Não existem dúvidas com relação à importância da energia elétrica, e, por isso, é preciso garantir que seu fornecimento ocorra normalmente. A matriz de energia brasileira é composta majoritariamente por fontes renováveis, mas é muito dependente da hidrelétrica. Essa dependência faz com que haja uma insegurança no fornecimento de energia em períodos de baixa precipitação pluviométrica, por exemplo, como já aconteceu anteriormente (Hocevar; Alves; Pereira, 2022).

Assim sendo, fica clara a necessidade de um planejamento para o fornecimento de energia elétrica. É importante buscar procedimentos e métodos de análise para compreender melhor padrões e tendências no consumo de eletricidade ao longo do tempo (Tidre; Biase; Silva, 2013).

A previsão da demanda de energia elétrica é uma ferramenta poderosa que consegue cumprir esse papel. Nesse contexto, faz-se necessária a previsão de uma série temporal.

2.2 Séries temporais

Séries temporais são dados extraídos de uma sequência de observações de uma grandeza ou um fenômeno registrados ao longo do tempo, geralmente com um intervalo regular entre amostras consecutivas. Via de regra, dados de uma série temporal têm uma relação intrincada e mútua, e, por isso, não podem ser considerados amostras

independentes. Conhecer tais relações e padrões presentes nas observações é de suma importância para o entendimento da série analisada.

Uma série de dados pode ser decomposta em quatro componentes: tendência (T_t), componente sazonal (S_t), componente cíclica e componente aleatória, sendo que as duas últimas são representadas em conjunto por a_t (Morettin; Toloi, 2006).

- **Tendência:** Define a dinâmica a longo prazo da série.
- **Componente sazonal:** descreve padrões que se repetem ao longo do tempo.
- **Componente cíclica:** descreve padrões que se repetem sem uma frequência definida, isto é, não são deterministicamente previsíveis.
- **Componente aleatória:** oscilações naturais imprevisíveis.

Existem métodos que, quando aplicados à série, podem diminuir a influência de algum dos componentes, melhorando a qualidade dos dados e, por sua vez, otimizando o processo de previsão.

A normalização é um método que tem como objetivo remover a componente sazonal dos dados. Sua aplicação se dá pela equação 1:

$$z_i = \frac{x_i - \hat{\mu}}{\hat{\sigma}} \tag{1}$$

Onde z_i é o dado normalizado, x_i é o dado sem normalização, $\hat{\mu}$ é a média dos dados e $\hat{\sigma}$ é a variância.

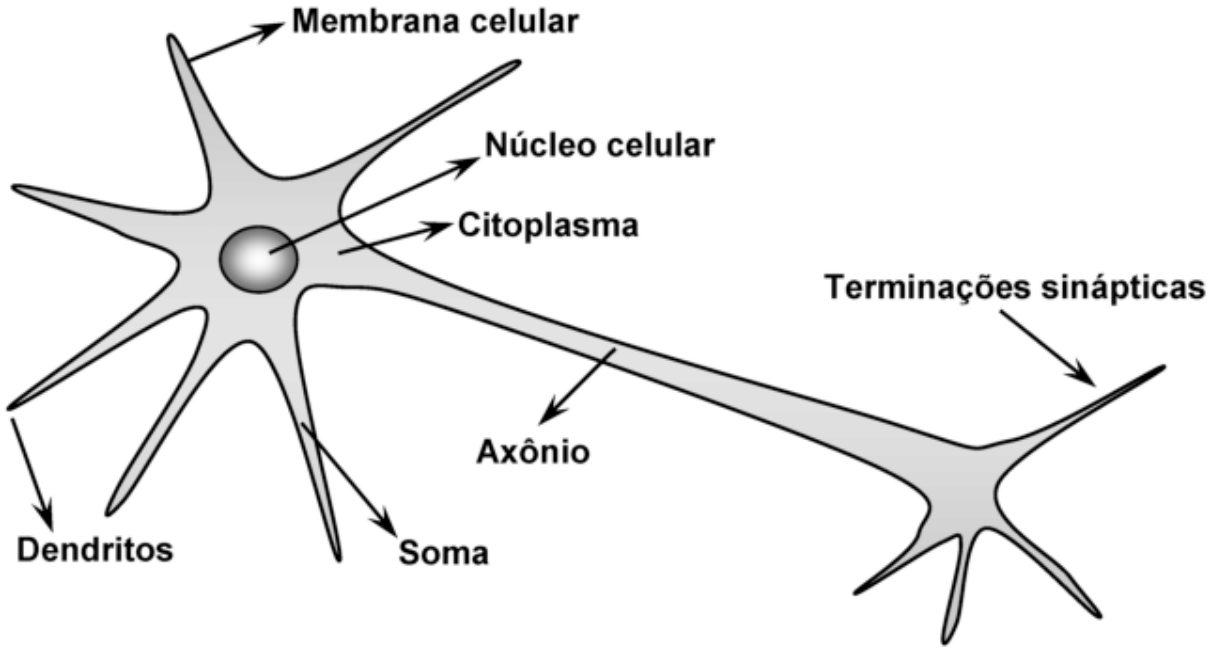
2.3 Inspiração biológica

A estrutura de processamento básica de uma Rede Neural Artificial (RNA) é o neurônio artificial. Ele foi criado inspirado no neurônio biológico (Haykin, 2008).

O neurônio biológico é a unidade de processamento básica do cérebro humano, sendo uma célula especializada na transmissão de informações (Silva; Spatti; Flauzino, 2010). Como é mostrado na figura 1, ele é dividido em três partes principais:

- **Dendritos** - São constituídos por vários finos prologamentos que formam a árvore dendrital. Têm como função captar estímulos/impulsos elétricos recebidos de outros neurônios.
- **Corpo celular** - Processa as informações recebidas dos dendritos e manda novos impulsos elétricos para o axônio.

Figura 1 – Representação de um neurônio biológico



Fonte: (Silva; Spatti; Flauzino, 2010)

- **Axônio** - Constituído por um único prolongamento que transmite os novos impulsos elétricos para os dendritos do neurônio seguinte. A sua terminação é constituída por ramificações chamadas terminações sinápticas.

O que viabiliza a transmissão de impulsos elétricos do axônio de um neurônio para os dendritos de outro são as terminações sinápticas. É pelas sinapses que os neurônios se conectam e formam uma rede neural.

2.4 Neurônio artificial

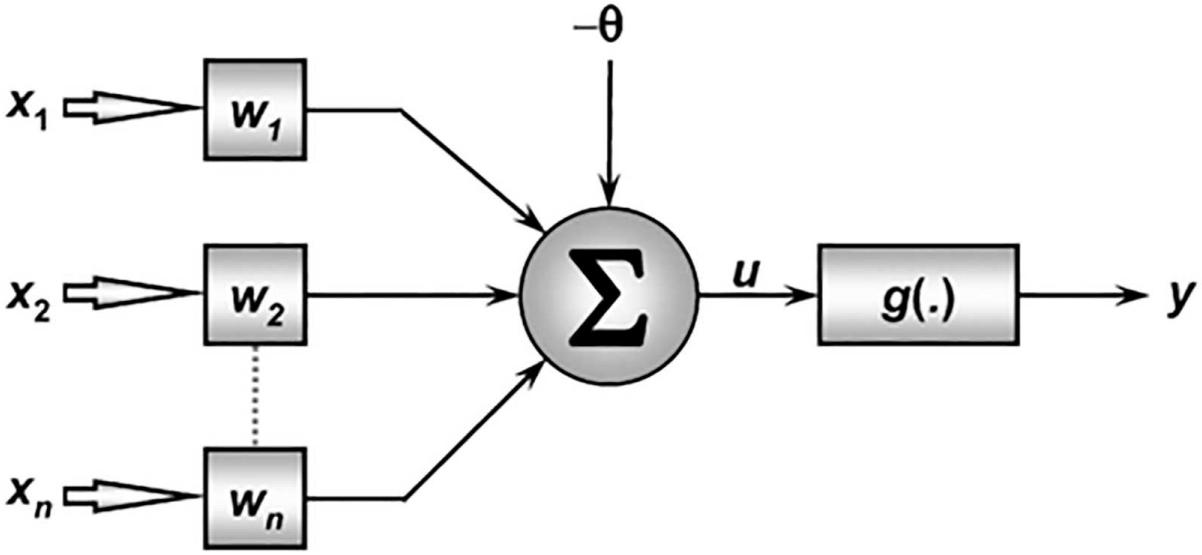
Inspirados no neurônio biológico, pesquisadores tentaram simular esse sistema em computador. O modelo mais bem aceito foi proposto por Warren McCulloch e Walter Pitts em 1943, e é, até hoje, o modelo mais utilizado nas diferentes arquiteturas de RNAs (Haykin, 2008).

Neste modelo de neurônio artificial, os impulsos elétricos vindos de outros neurônios são representados pelo conjunto de sinais de entrada $\{x_1, x_2, ..., x_n\}$. As ponderações feitas pelas junções sinápticas do modelo biológico são representadas pelo conjunto de pesos sinápticos $\{w_1, w_2, ..., w_n\}$. Dessa forma, a relevância de cada uma das entradas se dá por meio da multiplicação de cada uma das entradas $\{x_i\}$ com

seu respectivo peso sináptico $\{w_i\}$. Assim, a saída do corpo celular artificial, denotado por u, é a soma ponderada de suas entradas (Silva; Spatti; Flauzino, 2010).

A figura 2 contém uma representação do neurônio artificial. É possível afirmar que ele é constituído por sete elementos básicos:

Figura 2 – Representação de um neurônio artificial



Fonte: (Silva; Spatti; Flauzino, 2010)

- **1. Sinais de entrada {$x_1, x_2, ..., x_n$}**

Sinais ou medidas advindas do meio externo e que representam os valores assumidos pelas variáveis de uma aplicação específica. Esses sinais normalmente são normalizados a fim de aumentar a eficiência computacional do algoritmo de aprendizagem.

- **2. Pesos sinápticos {$w_1, w_2, ..., w_n$}**

Valores que servirão para ponderar cada uma das variáveis de entrada, quantificando a relevância da entrada em relação à funcionalidade do neurônio.

- **3. Combinador linear {Σ}**

Tem como função agregar todos os sinais de entrada já ponderados a fim de produzir um valor potencial de ativação.

- **4. Limiar de ativação (*bias*) {θ}**

Uma variável que especifica qual é o patamar apropriado para que o resultado produzido pelo combinador linear possa gerar um valor de disparo em direção à saída do neurônio.

- **5. Potencial de ativação {u}**

Resultado produzido pela diferença do valor produzido entre o combinador linear e o *bias*. Se $u \geq 0$, então o neurônio produz um potencial excitatório. Caso contrário, o neurônio produz um potencial inibitório.

- **6. Função de ativação {g(.)}**

Tem como objetivo limitar a saída do neurônio dentro de um intervalo de valores razoáveis a serem assumidos pela sua própria imagem funcional. Insere não-linearidade no sinal de saída, caso a função seja não-linear.

- **7. Sinal de saída {y}**

Valor final produzido pelo neurônio em relação a um determinado conjunto de sinais de entrada, podendo ser utilizado como conjunto de entrada por outros neurônios.

As equações 2 e 3 sintetizam o resultado produzido pelo neurônio artificial:

$$u = \sum_{i=1}^{n} w_i \times x_i - \theta \tag{2}$$

$$y = g(u) \tag{3}$$

De forma resumida, o neurônio artificial tem o seguinte funcionamento:

- Conjunto de sinais de entrada é multiplicado pelo conjunto de pesos sinápticos;
- Entradas ponderadas são somadas e o *bias* é subtraído, obtendo-se o potencial de ativação;
- Aplicação de uma função de ativação apropriada ao potencial de ativação;
- Compilação da saída.

2.4.0.1 Função de ativação

As funções de ativação mais utilizadas são (LeCun; Bengio; Hinton, 2015):

- **Sigmoide**

A função sigmoide (equação 4) pode ser interpretada como uma função degrau suavizada. Ela compreende o intervalo [0, 1] e sua derivada é dada pela equação 5.

$$\sigma(x) = (1 + e^{-x})^{-1} \tag{4}$$

$$\sigma'(x) = \sigma(x)(1 - \sigma(x)) \tag{5}$$

- **Tangente hiperbólica**

A função tangente hiperbólica (equação 6) tem um formato parecido com a sigmoide, com a diferença que compreende o intervalo [-1, 1]. Sua derivada é dada pela equação 7.

$$tanh(x) = \frac{1 - e^{-2x}}{1 + e^{-2x}} \tag{6}$$

$$tanh'(x) = 1 - tanh^2(x) \tag{7}$$

- **Rectified Linear Unit (ReLU)**

A função ReLU (equação 8) assume o valor 0 para entradas negativas e, para entradas positivas, assume o valor da função identidade. Sua derivada é dada pela equação 9. A função ReLU é uma das mais utilizadas em RNAs, tanto pela simplicidade quanto pela característica de não saturação, diferente das funções sigmoide e tangente hiperbólica (Géron, 2019).

$$ReLu(x) = \begin{cases} 0 & \text{se } x < 0 \\ x & \text{se } x \geq 0 \end{cases} \tag{8}$$

$$ReLu'(x) = \begin{cases} 0 & \text{se } x < 0 \\ 1 & \text{se } x \geq 0 \end{cases} \tag{9}$$

2.5 Redes Neurais Artificiais

Redes Neurais Artificiais são modelos matemáticos adaptativos e não lineares inspirados no sistema nervoso de seres vivos. Elas possuem a capacidade de aquisição e manutenção do conhecimento e podem ser definidas como um conjunto de neurônios artificiais interconectados. A maneira como os neurônios estão arranjados, a arquitetura da RNA, porém, varia muito (Silva; Spatti; Flauzino, 2010).

Uma RNA pode ser dividida em três partes, chamadas camadas, da seguinte forma:

- **Camada de entrada**

Camada responsável pelo recebimento de informações (dados), sinais, características ou medições advindas do meio externo. Normalmente, estas entradas são normalizadas em relação às faixas de variações dinâmicas produzidas pelas funções de ativação a fim de uma melhor precisão numérica com relação às operações matemáticas realizadas pela rede.

- **Camadas ocultas**

Camadas (ou camada, pode ser uma ou várias) compostas de neurônios com a capacidade de extrair as características associadas ao processo ou sistema a ser inferido. Quase todo o processo interno da rede acontece aqui.

- **Camada de saída**

Camada também composta de neurônios, com o objetivo de produzir e apresentar os resultados finais da rede.

Além da divisão em camadas, existem outras características comuns a diferentes arquiteturas de RNAs.

2.5.1 Características das redes neurais

Existem três classificações relevantes das diferentes arquiteturas das RNAs: quanto ao processo de treinamento, quanto à presença ou não de laços de aprendizado e quanto à natureza da função de ativação utilizada nos neurônios.

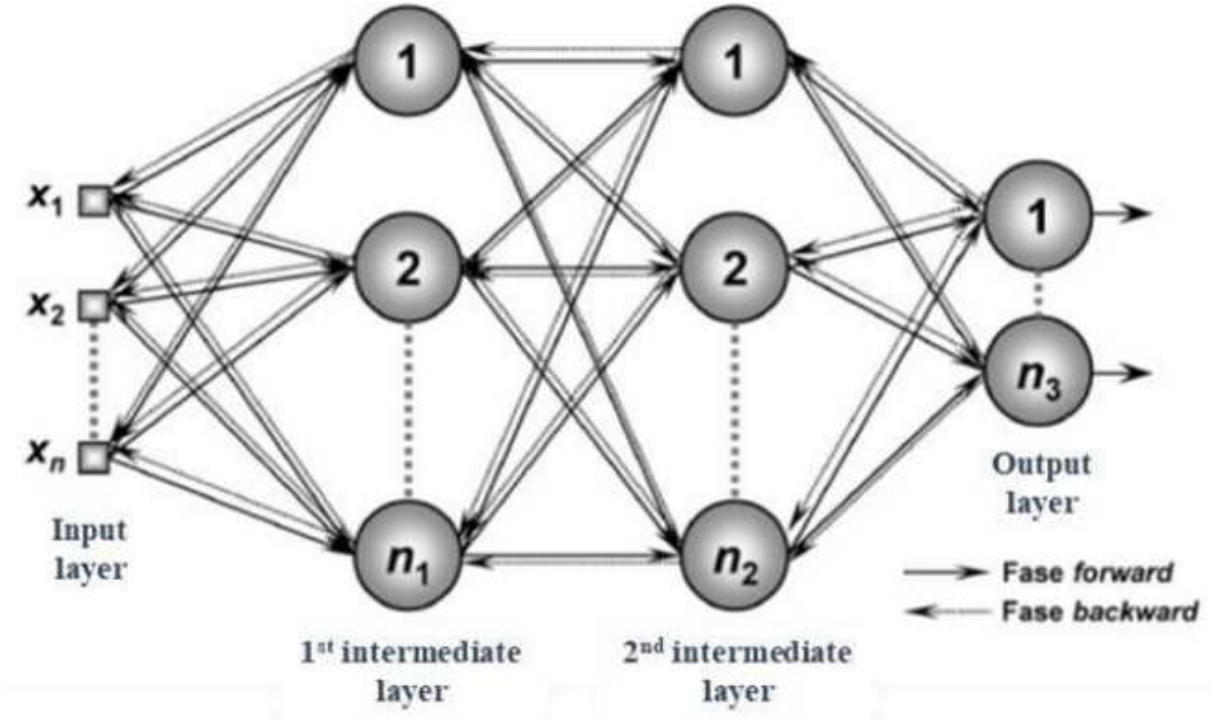
2.5.1.1 Redes recorrentes e não recorrentes

Uma classificação estrutural importante das RNAs diz como as camadas delas são interconectadas. As RNAs podem ser recorrentes ou não recorrentes (Haykin, 2008).

A classe não recorrente (*feedforward*, em inglês) se refere às redes em que a interconexão entre as camadas acontece em apenas um sentido, ou seja, é unidirecional. Esse sentido sempre se dá da camada de entrada, passando pelas camadas ocultas, até a camada de saída.

A figura 3 mostra uma rede não recorrente composta por uma camada de entrada com n sinais de entrada, duas camadas ocultas com n_1 e n_2 neurônios, respectivamente, e uma camada de saída com m neurônios. A quantidade de sinais de saída depende da quantidade de neurônios na camada de saída.

Figura 3 – Estrutura de rede *feedforward*



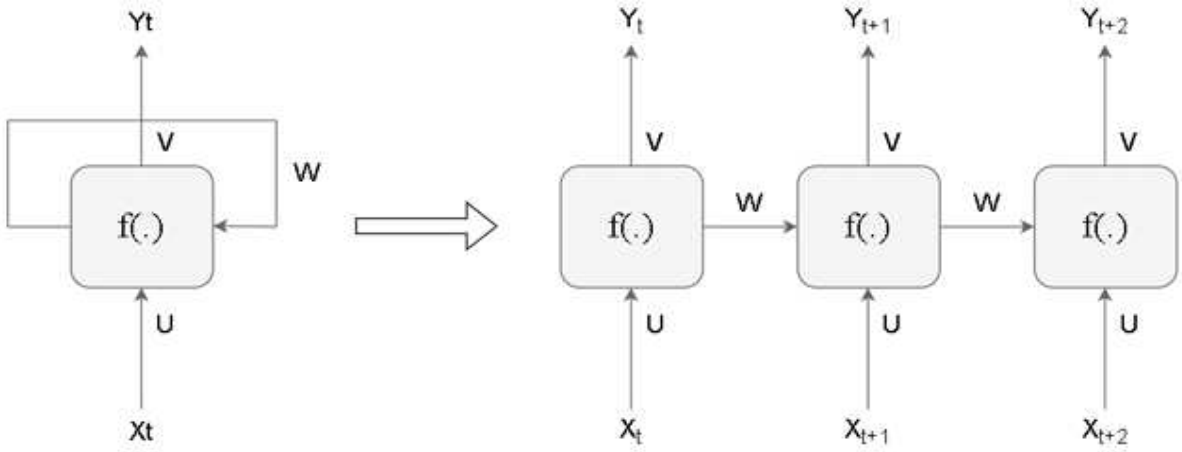
Fonte: (Rivero; Junior; Corrêa, 2023)

Uma característica das redes não recorrentes é que todos os neurônios de uma camada se conectam com todos os neurônios da camada seguinte, mas não se conectam com nenhum da mesma camada. Por exemplo, na figura 3, o neurônio 1 da primeira camada oculta se conecta com todos os neurônios da segunda camada oculta, mas não se conecta com nenhum outro neurônio da primeira camada oculta.

O treinamento dessa classe se dá através do método do gradiente descendente com o suporte do método de retropropagação de erro (*backpropagation*).

Já a classe recorrente (*feedback*, em inglês) se refere a redes que possuem realimentação de saídas da rede ou de camadas internas. Nesse caso, devido ao mecanismo de realimentação, para o treinamento, é necessário utilizar uma técnica de *backpropagation through time* (BPTT), na qual a rede é desdobrada temporalmente a cada época, formando uma sequência de redes não recorrentes iguais conectadas entre si. Utilizando o BPTT, basta utilizar um algoritmo de gradiente descendente. A figura 4 demonstra o funcionamento de uma rede recorrente desdobrada em três épocas pela técnica de BPTT.

Figura 4 – RNA desdobrada em três épocas pela técnica de BPTT.



Fonte: (Coimbra, 2023)

2.5.1.2 Mecanismos de treinamento

Uma das principais características das RNAs é a sua capacidade de aprender a partir da apresentação de padrões. Assim, após a rede ter aprendido a relação entre entradas e saídas, ela consegue generalizar uma solução. Em outras palavras, depois de treinada, a rede adquire a capacidade de produzir saídas satisfatórias para qualquer conjunto de entrada (Silva *et al.*, 2017).

Os passos para o treinamento de uma rede são chamados de algoritmo de treinamento. Esse processo consiste, basicamente, em otimizar os pesos sinápticos e o *bias* dos neurônios para que a rede atinja uma boa capacidade de generalização.

Os dados da série a ser prevista são divididos em dois conjuntos: conjunto de treinamento e conjunto de teste. Os dados de treinamento serão utilizados para

alimentar a rede na fase de treinamento. Já os de teste serão utilizados para verificar se a capacidade de generalização da rede está aceitável.

Cada vez que todos os dados de treinamento são passados para a rede e os pesos sinápticos são ajustados, acontece uma época de treinamento.

- **Treinamento supervisionado**

Durante a fase de treinamento, os ajustes dos pesos são feitos com um valor de referência. Dessa maneira, é necessário que cada dado de entrada tenha um dado de saída esperado. Assim, a cada época de treinamento, a resposta da rede a uma entrada é comparada com o seu valor esperado, resultando em um valor de erro.

Um algoritmo de otimização é utilizado para reduzir cada vez mais esse erro. A rede é considerada treinada quando o valor de erro é considerado aceitável, levando em consideração a capacidade de generalização.

- **Treinamento não supervisionado**

Diferente do método supervisionado, o próprio algoritmo busca encontrar padrões desconhecidos em dados não rotulados, isto é, dados de entrada que não têm valores esperados para a saída.

- **Treinamento por reforço**

Assim como no treinamento supervisionado, aqui é necessário que cada dado de entrada tenha um dado de saída esperado. O algoritmo baseia-se em um ou mais agentes que buscam pela melhor ação a ser tomada a fim de atingir o maior valor de recompensa em determinada situação. O algoritmo aprende por tentativa e erro até encontrar a melhor solução. A recompensa é um valor mensurável que recebe de "*feedback*"ao realizar determinada ação ou conjunto de ações.

Para a tarefa de previsão, geralmente é utilizado o algoritmo de treinamento supervisionado. O treinamento não supervisionado é mais utilizado para tarefas de classificação, onde o algoritmo, ao buscar padrões nos dados, encontra similaridades e diferenças entre eles e os agrupa, fazendo, assim, uma *clusterização*. Já o treinamento por reforço é uma técnica mais recente, com funcionamento complexo e não ideal quando é necessária alta eficiência computacional.

2.5.1.3 Validação

A busca de hiperparâmetros como número de camadas ocultas, número de neurônios por camadas, etc. não pode ser feita com o conjunto de teste, pois, assim, os hiperparâmetros produziriam o melhor modelo para esse conjunto em particular, e a rede não ficaria com uma boa capacidade de generalização. Para evitar isso, existe uma técnica de busca de hiperparâmetros chamada validação (Géron, 2019).

Nesta técnica, os dados de treinamento são divididos novamente, formando, assim, um conjunto de treino para realizar o treinamento dos hiperparâmetros e um conjunto de validação para fazer a avaliação dos hiperparâmetros escolhidos. É importante ressaltar que os dados de validação são diferentes dos dados de teste, já que o primeiro é utilizado para a escolha dos hiperparâmetros e o segundo é usado para avaliar o modelo final.

Existem diferentes técnicas de validação, sendo a validação *hold out* uma das principais. Nela, a série é dividida nos conjuntos de treinamento, validação e teste, sendo 70%, 15% e 15% dos dados para cada conjunto, respectivamente.

A rede é treinada com o conjunto de treinamento para cada combinação de hiperparâmetros, e o erro de cada combinação é encontrado utilizando o conjunto de validação. Assim, os hiperparâmetros são definidos como a combinação que obteve o menor erro de validação. A rede, então, é re-treinada e validada com o conjunto de teste.

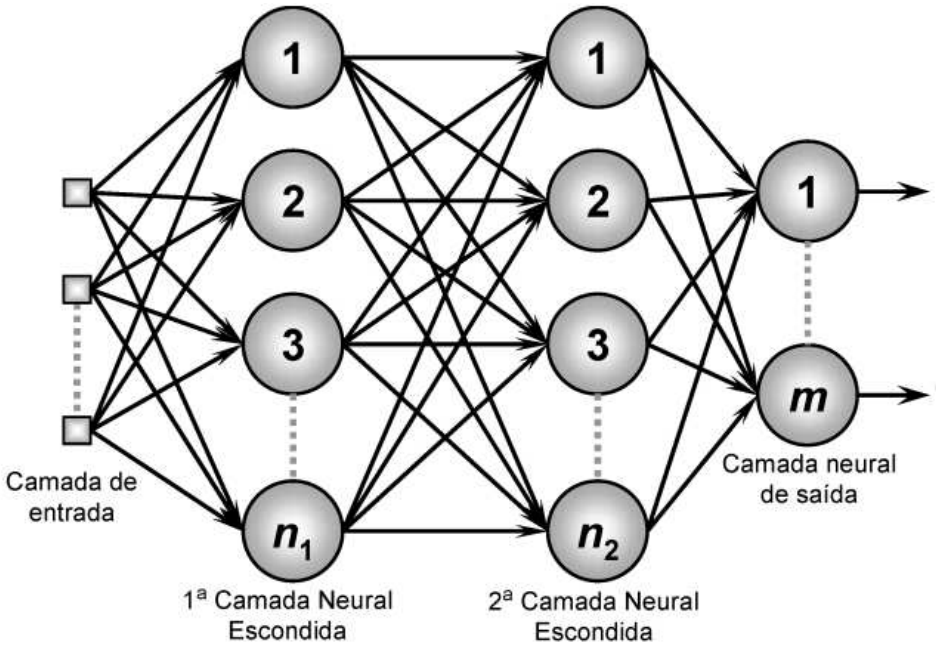
2.5.2 Perceptron de Múltiplas Camadas (MLP)

As redes Perceptron de Múltiplas Camadas (MLP, do inglês *Multilayer Perceptron*) são caracterizadas pela ampla variedade de aplicação em problemas de diversas áreas do conhecimento, sendo consideradas uma das arquiteturas mais versáteis em relação à aplicabilidade (Silva; Spatti; Flauzino, 2010). Entre essas áreas, algumas são:

- Aproximação universal das funções;
- Reconhecimento de padrões;
- Identificação e controle de processos;
- Previsão de séries temporais;
- Otimização de sistemas.

A rede MLP tem uma arquitetura não recorrente e seu treinamento ocorre de maneira supervisionada. A figura 5 deixa claro que o fluxo de informações se inicia na camada de entrada, percorre as camadas ocultas (ou intermediárias) e termina na camada de saída.

Figura 5 – Arquitetura da rede MLP



Fonte: (Silva; Spatti; Flauzino, 2010)

A especificação da configuração topológica de uma MLP, como a quantidade de camadas ocultas e seus respectivos números de neurônios, depende da quantidade de dados e sua complexidade. Já o ajuste dos pesos sinápticos de cada neurônio é efetuado durante o treinamento.

O treinamento de uma MLP pode ser entendido como uma tarefa de otimização não-linear, com a minimização de uma função custo baseada em erro de aproximação. O método mais básico utilizado para isso é o gradiente descendente. Junto a ele, também é necessária a utilização do algoritmo de retropropagação de erro (*backpropagation*).

O algoritmo *backpropagation* é dividido em duas fases. A primeira fase a ser aplicada é denominada *forward*, na qual os dados de entrada $\{x_1, x_2, ..., x_n\}$ do conjunto de dados de treinamento são inseridos nas entradas da rede e são propagados camada a camada até a produção da respectiva saída. Esta saída provavelmente não será satisfatória, visto que os pesos são inicializados aleatoriamente. Então, é calculado o erro instantâneo e entre a saída y produzida pelo neurônio j da camada de saída na iteração n e a sua saída desejada d. Os cálculos da saída da rede e o erro obtido são

dados pelas equações 10 e 11, respectivamente:

$$y_j(n) = \varphi_j(v_j(n)) \tag{10}$$

Sendo ϕ_j a função de ativação do neurônio da camada de saída e v_j os dados vindos da última camada oculta.

$$e_j(n) = d_j(n) - y_j(n) \tag{11}$$

Ocorre, então, a segunda fase da *backpropagation*, a fase *backward*. O erro calculado é utilizado para alterar os valores dos pesos dos neurônios pelo método do gradiente descendente. São calculados os gradientes locais da rede (δ) de acordo com a equação 12:

$$\delta_j^{(l)}(n) = \begin{cases} e_j^{(L)}(n)\varphi_j'(v_j^{(L)}(n)) & \text{para neurônio } j \text{ na camada de saída } L \\ \varphi_j'(v_j^{(l)}(n))\sum_k \delta_k^{(l+1)}(n) w_{kj}^{(l+1)}(n) & \text{para neurônio } j \text{ na camada oculta } l \end{cases} \tag{12}$$

Então, a equação 13 é aplicada para atualizar os vetores de pesos:

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \eta\delta_j^{(l)}(n) y_i^{(l-1)}(n) \tag{13}$$

Sendo $w_{ji}^{(l)}$ o vetor de pesos, η o passo de aprendizagem e $y_i^{(l-1)}$ a saída da camada anterior.

Dessa forma, aplicações sucessivas das fases *forward* e *backward* fazem com que os pesos sinápticos dos neurônios se ajustem automaticamente a cada época de treinamento, implicando numa gradativa diminuição da soma dos erros produzidos pelas respostas da rede em relação às saídas desejadas.

2.5.3 Função de Base Radial (RBF)

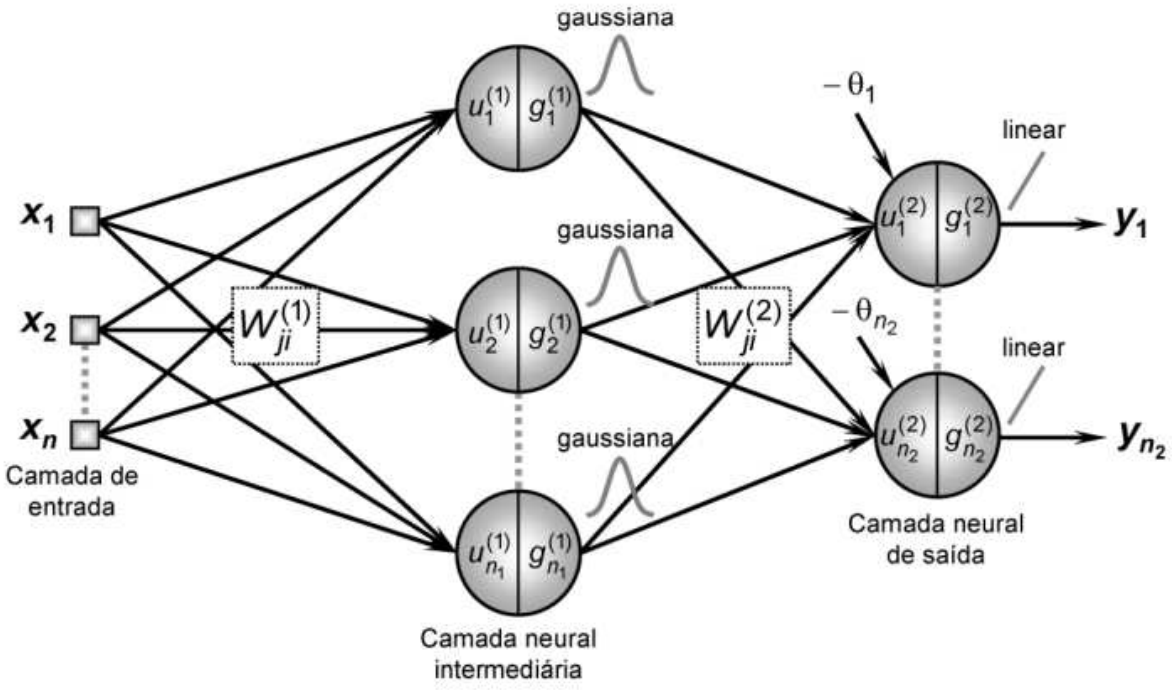
As redes de Funções de Base Radial (RBF, do inglês *Radial Basis Function*) são muito semelhantes às redes MLP e podem ser empregadas em quase todos os mesmos problemas tratados por elas (Silva; Spatti; Flauzino, 2010).

A rede RBF também pertence à arquitetura não recorrente e tem treinamento supervisionado. O fluxo de informações na sua estrutura também se inicia na camada de entrada, passa pela camada oculta e termina na camada de saída.

Porém, diferente das redes MLP, que podem ter várias camadas ocultas, a estrutura típica de uma RBF tem apenas uma, e seus neurônios têm funções de ativação de base radial, como as gaussianas (Haykin, 2008).

A figura 6 mostra a arquitetura da rede RBF. É possível notar a camada de entrada, a camada oculta única com neurônios com função de ativação gaussiana e a camada de saída com neurônios com função de ativação linear.

Figura 6 – Arquitetura da rede RBF



Fonte: (Silva; Spatti; Flauzino, 2010)

O processo de treinamento da RBF também difere bastante da MLP. Ele é constituído de duas fases. A primeira consiste em encontrar os centros e raios dos neurônios da camada intermediária por meio de um método de aprendizagem não supervisionado, dependente apenas das características dos dados de entrada. Este ajuste está diretamente relacionado ao uso das funções de base radial. A segunda fase consiste em um ajuste dos pesos dos neurônios da camada de saída, utilizando, assim como o MLP, o gradiente descendente.

Funções de bases radiais são funções que, como o próprio nome diz, têm uma base circular ou elipsoidal. A primeira fase de treinamento da RBF consiste em utilizar

um algoritmo para achar o centro e o raio dessas bases (Haykin, 2008).

Um algoritmo muito utilizado para este objetivo é o *k-means* (*k*-médias), cujo propósito é posicionar os centros de *k*-gaussianas em regiões onde os padrões de entrada tenderão a se agrupar.

Após definir o número de neurônios da camada oculta, que corresponde ao parâmetro *k*, o algoritmo inicializa o vetor de peso de cada neurônio de forma aleatória. Após isso, são calculadas as distâncias euclidianas de cada um dos dados de treinamento em relação aos neurônios, e cada amostra será inserida no conjunto do neurônio que estiver mais próximo de si.

Em seguida, é recalculada uma nova posição para os centros e os vetores de peso W são atualizados de acordo com a equação 14:

$$W_{ji} = \frac{1}{m^{(j)}} \times \sum_{x^{(k)} \in \Omega^{(j)}} x^{(k)} \tag{14}$$

Onde $x^{(k)}$ são os dados de treinamento, $\Omega^{(j)}$ são os conjuntos dos neurônios e $m^{(j)}$ é o número de amostras em $\Omega^{(j)}$.

Estes passos são repetidos de maneira iterativa até que seja atingido um critério de parada ou que os centros não mudem mais de posição. Então é calculada a variância σ de cada uma das funções gaussianas de acordo com a equação 15:

$$\sigma = \frac{d_{max}}{\sqrt{2m^{(j)}}} \tag{15}$$

Sendo d_{max} a distância máxima entre os centros.

Sendo treinada a camada oculta, falta apenas treinar a camada de saída. Para isso, é preciso computar a saída h da camada oculta e aplicar a equação 16:

$$y_j(n) = \varphi_j(h_j(n)) \tag{16}$$

É necessário, então, calcular o erro de saída e os gradientes locais da rede, de acordo com as equações 17 e 18, respectivamente:

$$e_j(n) = d_j(n) - y_j(n) \tag{17}$$

$$\delta_j^{(l)}(n) = e_j^{(L)}(n)\varphi_j'(h_j^{(L)}(n)) \tag{18}$$

Por fim, os pesos da camada de saída são gerados aleatoriamente. Para atualizá-los, é utilizada a equação 19:

$$w_{ji}^{(l)}(n+1) = w_{ji}^{(l)}(n) + \eta\delta_j^{(l)}(n)y_i^{(l-1)}(n) \tag{19}$$

2.5.4 Máquinas de Aprendizado Extremo (ELM)

As Máquinas de Aprendizado Extremo (ELM, do inglês, *Extreme Learning Machine*) são redes neurais também semelhantes às MLPs, sendo não recorrentes e tendo treinamento supervisionado, mas possuindo apenas uma camada intermediária (Huang; Zhu; Siew, 2004).

Uma diferença notável entre as redes é que na ELM não é necessário definir uma função de otimização, pois seu treinamento utiliza a pseudo inversa de Moore-Penrose, que garante a norma mínima global dos pesos sinápticos, maximizando, assim, a capacidade de generalização da rede.

A figura 7 apresenta a arquitetura da rede ELM. É possível notar a ausência da camada de entrada: os dados de entrada $\{x_1, x_2, ..., x_n\}$ são passados diretamente à camada oculta, onde são multiplicados pelo vetor W^h que contém os pesos gerados de forma aleatória. O sinal produzido por essa camada se dá pela equação 20:

$$x_t^h = f^h(W^h u_t + b) \tag{20}$$

Onde $f^h(.)$ é a função de ativação dos neurônios e b é o limiar de ativação.

x_t^h é, então, passado para a camada de saída, onde ocorre a combinação linear das ativações dos neurônios da camada oculta de acordo com a equação 21:

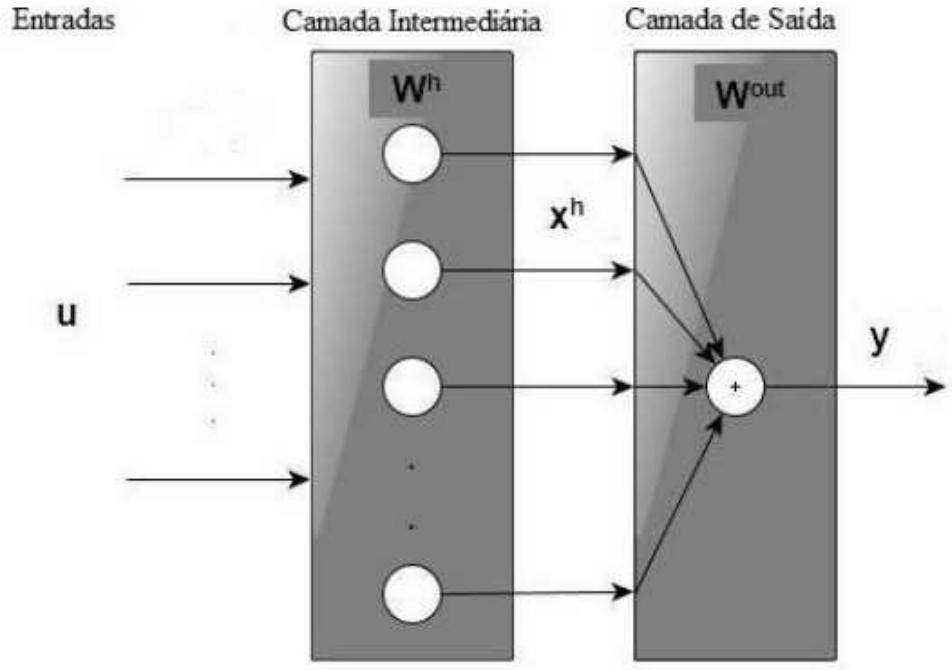
$$y_t = W^{out}x_t^h \tag{21}$$

Onde y_t é a saída da rede e W^{out} são os pesos sinápticos da camada de saída.

O treinamento da ELM é muito mais simples do que o da MLP. Os pesos dos neurônios da camada oculta são definidos aleatoriamente e não são alterados em momento algum. Dessa forma, o processo de ajuste adapta apenas os pesos da camada de saída.

Esse ajuste se dá pela resolução do operador Moore-Penrose, representada na equação 22:

Figura 7 – Arquitetura da rede ELM



Fonte: (Siqueira, 2013)

$$W^{out} = (X_h^t X_h)^{-1} X_h^t d \tag{22}$$

Onde X_h é a matriz de saídas da camada oculta, $(X_h^t X_h)^{-1} X_h^t$ é a pseudo-inversa de X_h e d são os dados de saída desejados.

Para aumentar o desempenho da rede, é possível implementar um processo de validação do treinamento utilizando um coeficiente de regularização C. Este coeficiente é adicionado de acordo com a equação 23:

$$W^{out} = (\frac{1}{C} + X_h^t X_h)^{-1} X_h^t d \tag{23}$$

Para esse processo, é utilizado um conjunto de dados de validação. Esses dados são apresentados à rede assumindo que $C = 2^{\lambda}$, sendo que $\lambda = \{2^{-25}, 2^{-24},...,2^{25}, 2^{26}\}$. O coeficiente C que gera o menor erro de validação é considerado o que permite à rede a melhor generalização.

2.5.5 Rede com Estado de Eco (ESN)

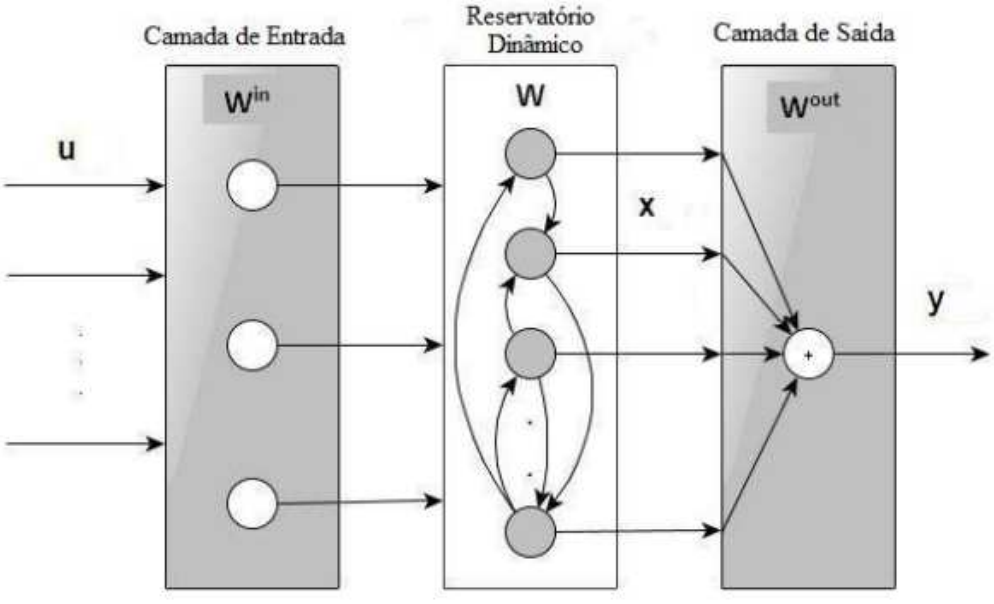
Redes não recorrentes clássicas têm um problema com relação à previsão de séries temporais: elas não levam em consideração o caráter sequencial da série. Redes

recorrentes, por sua vez, possuem laços de realimentação entre neurônios e, por isso, conseguem armazenar informações sequenciais. Por outro lado, a realimentação faz com que o treinamento seja muito mais complexo e a aplicação de métodos de otimização, por exemplo, pode levar a configurações instáveis. Dessa forma, as vantagens dessas redes ficam limitadas pela dificuldade de ajuste dos seus parâmetros livres.

A Rede com Estado de Eco (ESN, do inglês *Echo State Network*), no entanto, é uma rede recorrente com um treinamento simples, sem os problemas citados anteriormente. Sua camada intermediária, chamada de reservatório de dinâmicas, é composta por neurônios não lineares totalmente interconectados, e seus pesos sinápticos são definidos aleatoriamente e inalterados. Os sinais saídos do reservatório são chamados de estado de eco e são passados para a camada de saída, que produz as saídas utilizando uma combinação linear (Jaeger, 2001). Dessa maneira, o seu treinamento, assim como o da ELM, ajusta apenas os pesos sinápticos dos neurônios da camada de saída.

A figura 8 apresenta a arquitetura de uma rede ESN. Nela, as entradas são representadas pelo vetor $u_t = [u_1, u_2, ..., u_{t-K+1}]$, sendo K o número de entradas da rede. Elas são ponderadas linearmente pelos coeficientes W^{in} da camada de entrada e passam ao reservatório de dinâmicas W. No reservatório, são gerados os estados de eco $x_t = [x_t^1, x_t^2, ..., x_t^N]^T$ que são atualizados de acordo com a equação 24:

Figura 8 – Arquitetura da rede ESN



Fonte: (Siqueira, 2013)

$$x_{t+1} = f(w^{in}u_{t+1} + Wx_t) \tag{24}$$

Onde $f(.) = (f_1(.), f_2(.),...,f_N(.)$ representa as ativações dos neurônios do reservatório, sendo N o número de neurônios do reservatório. É importante pontuar que a inicialização dos estados desta rede é feita com o valor nulo. Por último, a saída da rede é dada pela equação 25:

$$y_{t+1} = W^{out}x_{t+1} \tag{25}$$

Sendo W^{out} a matriz contendo os pesos da camada de saída.

Para definir W^{out}, assim como na ELM, é possível utilizar o operador de Moore-Penrose a partir dos estados de eco X e das saídas desejadas d, como mostrado na equação 26:

$$W^{out} = (X^tX)^{-1}X^td \tag{26}$$

É preciso salientar que a matriz W deve obedecer às propriedades de estado de eco, que são, basicamente, condições que, ao serem seguidas, garantem que a rede tenha estados de eco. A primeira propriedade diz que se o módulo do máximo valor singular da matriz estiver dentro do círculo real unitário, ou seja, $(|\sigma_{max}(W)|) < 1$, a rede apresentará estados de eco. Esta propriedade é aplicável caso a rede tenha neurônios com funções de ativação do tipo tangente hiperbólica no reservatório e não apresente realimentações da camada de saída para o reservatório (Jaeger, 2001).

A segunda propriedade diz que o maior autovalor em módulo da matriz de pesos, chamado de raio espectral, deve ser menor do que 1, ou seja, $|r_{max}(W)|) < 1$, para que a rede apresente estados de eco (Jaeger, 2001).

Assim, nota-se que é necessário definir W de forma a obedecer às propriedades de estado de eco, definir W^{in} de forma arbitrária e treinar a camada de saída conforme a equação 25.

Uma possibilidade de criação para a matriz W foi proposta por (Ozturk; Xu; Príncipe, 2007). Os autores propuseram uma estratégia na qual os autovalores respeitam uma distribuição uniforme no círculo unitário, criando uma matriz canônica, como mostrado na equação 27:

$$\mathbf{W} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 & -r^N \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{bmatrix} \tag{27}$$

Sendo r o raio espectral.

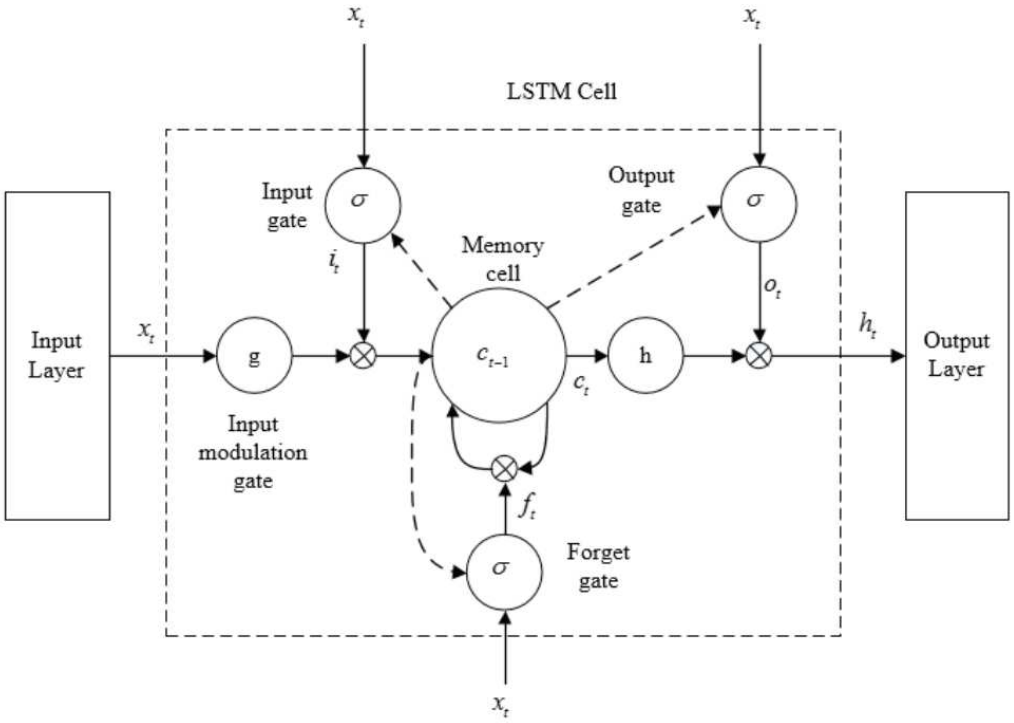
A ideia dessa proposta é que, na ausência de informações *a priori* sobre a saída desejada, é razoável espalhar de maneira uniforme estes autovalores na tentativa de buscar boas aproximações para quaisquer mapeamentos arbitrários.

2.5.6 Memória de Longo-Curto Prazo (LSTM)

A rede neural Memória de Longo-Curto Prazo (LSTM, do inglês *Long-Short Term Memory*) é um tipo de rede recorrente projetada para armazenar tanto termos de curto prazo, como a ELM, quanto termos de longo prazo, sem deixar que ocorram os problemas das redes recorrentes clássicas citados anteriormente. Ela é formada por células que contêm três portões (ou *gates*): *input gate*, *output gate* e *forget gate*. Cada célula guarda valores de intervalo de tempo arbitrários e os portões controlam o fluxo de informações. Os componentes básicos de uma LSTM (que podem ser vistos na figura 9) são (Hochreiter; Schmidhuber, 1997):

- *Cell State* (c_t): define o estado interno da célula, no qual são armazenadas e carregadas informações de longo prazo.
- *Hidden State* (h_t): memória de trabalho que armazena informações de curto prazo relativa aos eventos imediatamente anteriores da célula. Diferentemente do *cell state*, ela é sobrescrita a cada passo.
- *Forget Gate* (f_t): responsável por regular quais informações do *cell state* serão descartadas. É formado por uma rede neural com função de ativação sigmoide atribuindo pesos de 0 a 1 às informações. Se sua saída for 0, a informação é esquecida pela *cell state*. Se for 1, é mantida. A saída f_t desse bloco é dada pela equação 28:

Figura 9 – Arquitetura da rede LSTM



Fonte: (Fu; Zhang; Li, 2016)

$$f_t = \sigma(W_f[x_t, h_{t-1}] + b_f) \tag{28}$$

Onde x_t são as novas informações de entrada, W_f são os pesos relativos à combinação da entrada com o *hidden state* e b_f é o limiar da rede neural.

- *Input Gate* (i_t), (g_t): portão responsável por determinar quais informações poderão entrar no *cell state*. É composto por duas redes neurais, uma que seleciona novas informações candidatas e outra que decide quais destas serão realmente atualizadas no *cell state*. Este portão é representado pelas equações 29 e 30:

$$i_t = \sigma(W_i[x_t, h_{t-1}] + b_i) \tag{29}$$

$$g_t = tanh(W_g[x_t, h_{t-1}] + b_g) \tag{30}$$

Sendo W_i e W_g os pesos de entrada e b_i e b_g os limiares de cada rede neural. A saída da *input gate* é a multiplicação elemento por elemento de i_t e g_t. Essa multiplicação é identificada pelo símbolo $\odot$.

- *Output Gate* (o_t): tem a função de filtrar os elementos do *cell state*, decidindo quais elementos irão para a saída. É formado por uma rede neural com função de ativação sigmoide, e as informações do *cell state* são mapeada em valores de -1 a 1 por uma função tangente hiperbólica. A saída deste portão é dada pela equação 31:

$$o_t = \sigma(W_o[x_t, h_{t-1}] + b_o) \tag{31}$$

  Sendo W_o os pesos da camada de saída.

Finalmente, o *hidden state* e o *cell state* são atualizados, respectivamente, pelas equações 32 e 33:

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \tag{32}$$

$$h_t = o_t \odot tanh(c_t) \tag{33}$$

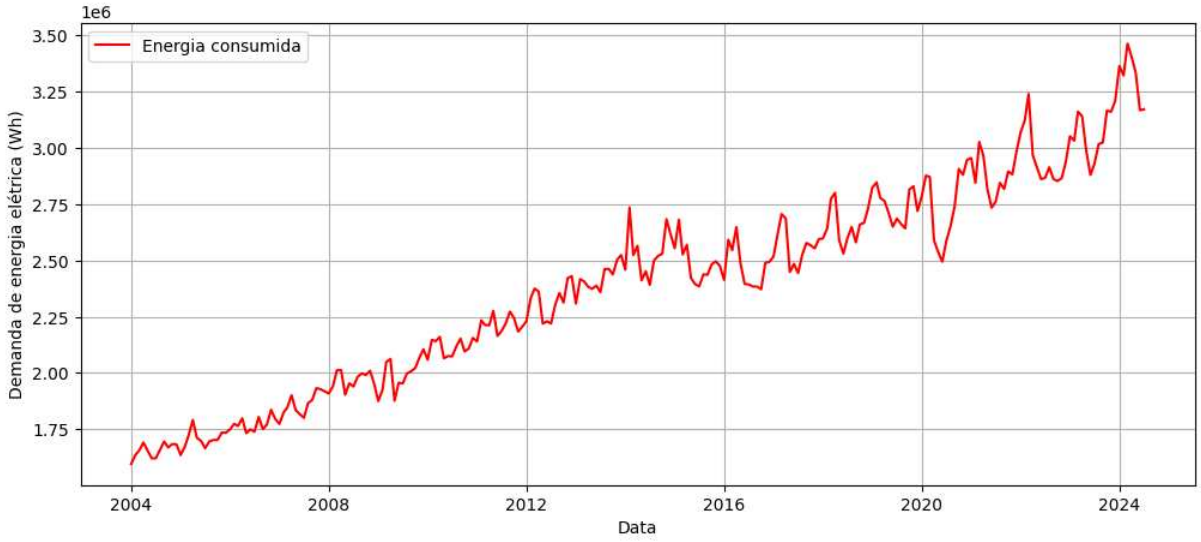
3 MATERIAL E MÉTODOS

3.1 Base de dados

O conjunto de dados (*dataset*) utilizado neste trabalho foi retirado do *site* da Empresa de Pesquisa Energética (EPE) (Energética, 2024). Ele contém 247 dados com o consumo mensal de energia elétrica, em MWh, dos estados do Paraná, de Santa Catarina e do Rio Grande do Sul, no período de janeiro de 2004 a julho de 2024.

Os gráficos 1, 2 e 3 exibem os dados das séries do Paraná, de Santa Catarina e do Rio Grande do Sul, respectivamente.

Gráfico 1 – *Dataset* Paraná



Fonte: autoria própria (2025)

É possível perceber que a demanda é fortemente influenciada pelas estações do ano, havendo um pico de consumo no verão, formando um padrão que se repete a cada doze meses. Isso confere às séries um caráter sazonal.

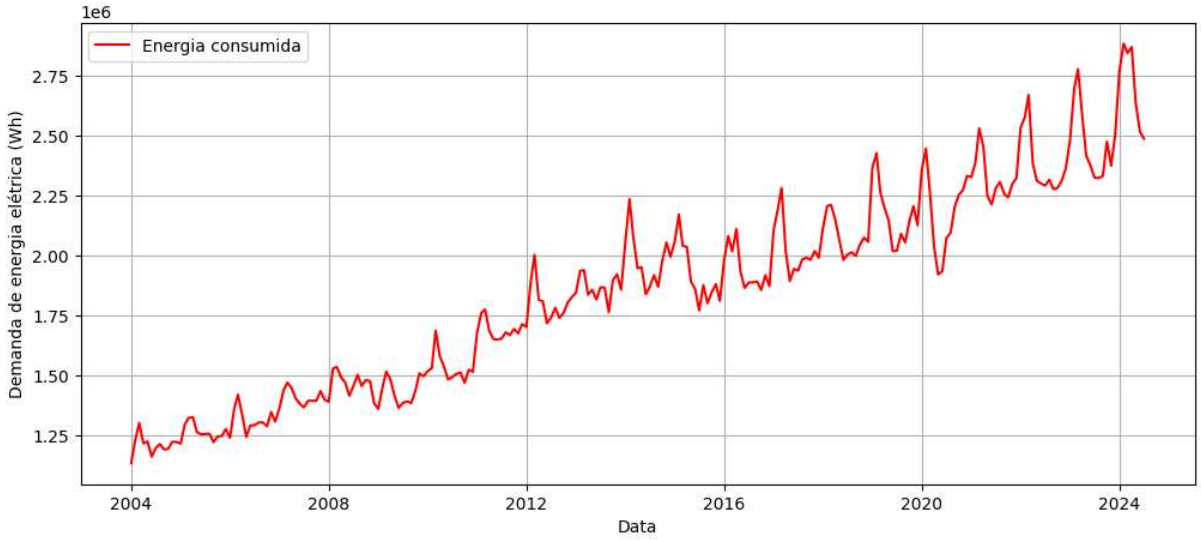
A tabela 1 exibe alguns dados de estatística descritiva das séries temporais utilizadas a fim de se ter uma melhor visão geral sobre cada uma delas.

Tabela 1 – Estatística descritiva das bases de dados

	Paraná	Santa Catarina	Rio Grande do Sul
Média	2381919.70	1848571.31	2318893.26
Mediana	2421205.56	1869948.00	2359020.00
Máximo	3462065.52	2886354.50	3058339.82
Mínimo	1596274.29	1132926.27	1621708.89
Desvio Padrão	449104.88	415633.97	312579.95

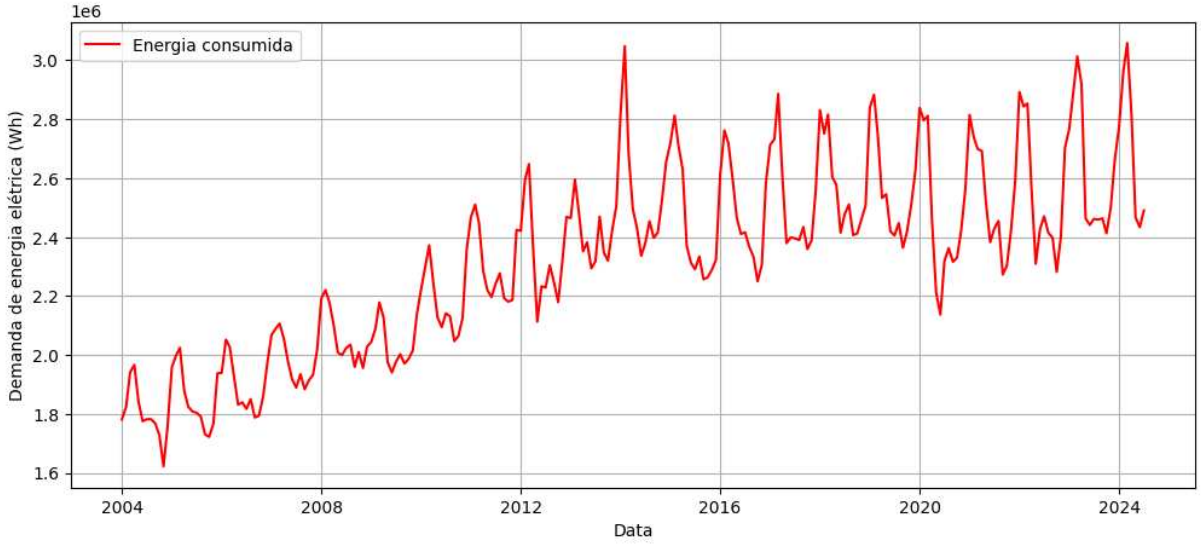
Fonte: autoria própria (2025)

Gráfico 2 – *Dataset* Santa Catarina



Fonte: autoria própria (2025)

Gráfico 3 – *Dataset* Rio Grande do Sul



Fonte: autoria própria (2025)

3.2 Pré-processamento

As séries foram divididas em três conjuntos: treinamento, contendo 75% dos dados, validação, contendo 15%, e teste, também contendo 15% dos dados. É importante ressaltar que essa divisão foi feita de maneira sequencial, com todos os conjuntos tendo dados consecutivos. Assim, sua relação intrincada é levada em consideração para a previsão.

Dessa forma, o conjunto de treinamento contém os dados de jan/2004 até jul/2018, o conjunto de validação contém os dados de ago/2018 até jul/2021 e, por último, o conjunto de teste contém os dados de ago/2021 até jul/2024.

Em seguida, os conjuntos foram normalizados separadamente, com o objetivo de remover a sazonalidade dos dados.

3.3 Ajustes dos hiperparâmetros

Foram testadas diferentes combinações de hiperparâmetros para cada rede neural. Foi utilizado o método de validação *hold out* para avaliar cada combinação no conjunto de validação, e a que obteve o menor erro de validação foi escolhida.

Primeiramente, foram determinados os parâmetros que não seriam varridos pela busca em grade. O número de camadas ocultas foi fixado em 1, tendo em vista que as séries temporais são pequenas, e foi utilizado 1 atraso da série como entrada para prever o próximo instante. A quantidade de épocas foi escolhida de maneira empírica, após vários testes realizados. Então, as funções de ativação foram escolhidas com base na literatura, sendo utilizadas funções com históricos de bons resultados. As funções definidas são mostradas na tabela 2:

Tabela 2 – Funções de ativação definidas

Rede	MLP	RBF	ELM	ESN	LSTM
Função de ativação	Tangente hiperbólica	Gaussiana	Sigmoide	X	X

Fonte: autoria própria (2025)

Para a MLP, a RBF, a ELM e a ESN, o número de neurônios na camada oculta foi escolhido dentre valores de 1 a 100. A quantidade foi mantida em valores razoavelmente pequenos devido ao tamanho da série temporal.

A ESN deve respeitar as propriedades de estado de eco, como mencionado anteriormente. Assim, foram testados diferentes valores para o raio espectral: 0,8; 0,85; 0,9; 0,95 e 0,99. As quantidades de reservatórios validadas foram 1 e 2.

Para a ELM, o coeficiente de regularização C foi definido como $C = 2^{\lambda}$, sendo que $\lambda = \{2^{-25}, 2^{-24}, ..., 2^{25}, 2^{26}\}$. O λ que gerou o menor erro de validação foi o escolhido.

A quantidade de blocos LSTM utilizados foi de apenas 1, por se tratar de séries temporais pequenas. A dimensão do estado interno foi testada com diferentes valores: 25, 50, 100, 150, 200, 250, 300 e 350.

3.4 Métricas

Após a obtenção dos dados previstos pelas redes, é necessário definir um critério de avaliação da qualidade da previsão do modelo estudado. Para isso, é possível utilizar alguma métrica de erro que calcule a diferença entre os dados previstos e os dados reais. No presente trabalho foram utilizados o Erro Quadrático Médio (MSE, do inglês *Mean Squared Error*), Raiz do Erro Quadrático Médio (RMSE, do inglês *Root Mean Squared Error*), o Erro Médio Absoluto (MAE, do inglês *Mean Absolute Error*) e o Erro Médio Percentual Absoluto (MAPE, do inglês *Mean Absolute Percentage Error*). Essas são algumas das métricas mais utilizadas para a avaliação de RNAs, todas com um grande histórico de utilização dentro da literatura (Harrison, 2019).

O MSE, comumente utilizado como função custo no treinamento de redes *feedforward*, calcula a média da diferença elevada ao quadrado entre o valor real e o valor previsto. Dessa forma, valores previstos muito diferentes dos reais são penalizados. O MSE é dado pela equação 34:

$$MSE = \frac{1}{N_s} \sum_{t=1}^{Ns} (X_t - \hat{X}_t)^2 \tag{34}$$

Sendo N_s o número de dados, X_t os dados reais e $\hat{X}_t$ os dados previstos.

O RMSE, por sua vez, é basicamente o mesmo cálculo do MSE, mas com a raiz quadrada aplicada. Dessa forma, seu resultado fica mais fácil de interpretar, já que ele tem a mesma unidade de medida dos dados utilizados. Sua forma é dada pela equação 35:

$$RMSE = \sqrt{\frac{1}{N_s} \sum_{t=1}^{Ns} (X_t - \hat{X}_t)^2} \tag{35}$$

O MAE também é muito parecido com o MSE, porém, a média da diferença entre o valor real e o valor previsto, ao invés de ser elevada ao quadrado, é colocada em módulo. Assim, esta métrica não é afetada por valores discrepantes (*outliers*). Sua fórmula é dada pela equação 36:

$$MAE = \frac{1}{N_s} \sum_{t=1}^{Ns} |X_t - \hat{X}_t| \tag{36}$$

Por último, o MAPE é o MAE dividido pelo módulo do valor real. Assim, o MAPE mostra a porcentagem de erro em relação ao valor real. Sua forma é dada pela equação 37:

$$MAPE = \frac{1}{N_s} \sum_{t=1}^{Ns} \left| \frac{X_t - \hat{X}_t}{X_t} \right| \tag{37}$$

Sendo N_s o número de dados, X_t os dados reais e $\hat{X}_t$ os dados previstos.

4 RESULTADOS E DISCUSSÃO

Como resultados das redes estudadas, têm-se os hiperparâmetros definidos para cada RNA, a avaliação da resposta obtida pelos modelos de previsão e a curva gerada pelos dados obtidos.

4.1 Hiperparâmetros finais

Após a realização da busca em grade e da observação dos erros de validação para cada combinação de hiperparâmetros testados, os parâmetros finais para cada RNA foram definidos. Eles estão dispostos nas tabelas 3, 4 e 5, que se referem às séries do Paraná, de Santa Catarina e do Rio Grande do Sul, respectivamente.

Tabela 3 – Hiperparâmetros finais - Paraná

	MLP	RBF	ELM	ESN	LSTM
N° de neurônios	100	2	3	2	200
N° de épocas	120	300	X	X	300
Raio espectral	X	X	X	0,95	X
Coeficiente C	X	X	2^3	X	X

Fonte: autoria própria (2025)

Tabela 4 – Hiperparâmetros finais - Santa Catarina

	MLP	RBF	ELM	ESN	LSTM
N° de neurônios	90	3	3	2	200
N° de épocas	130	200	X	X	250
Raio espectral	X	X	X	0,95	X
Coeficiente C	X	X	2^-12	X	X

Fonte: autoria própria (2025)

Tabela 5 – Hiperparâmetros finais - Rio Grande do Sul

	MLP	RBF	ELM	ESN	LSTM
N° de neurônios	100	5	3	3	200
N° de épocas	150	150	X	X	350
Raio espectral	X	X	X	0,99	X
Coeficiente C	X	X	2^5	X	X

Fonte: autoria própria (2025)

4.2 Avaliação das previsões

A fim de fazer uma avaliação dos erros de previsão, foram montadas tabelas que exibem os valores de erro de cada modelo desenvolvido para cada uma das séries temporais. Os erros mostrados nestas tabelas foram os menores dentre 30 simulações feitas com cada rede com seus hiperparâmetros finais. Para complementar a análise dos resultados obtidos, também foram plotados gráficos *boxplots* referentes às mesmas 30 simulações. Cada caixa é delimitada pelos quartis de 25%, 50% e 75%, e os valores máximos e mínimos são indicados pelas linhas externas às caixas.

4.2.1 Paraná

É possível perceber que na série temporal do estado do Paraná, a rede que obteve melhor desempenho foi a ESN, com os menores valores de RMSE, MSE, MAE e MAPE, enquanto a LSTM foi a menos efetiva.

Tabela 6 – Erros - Paraná

	MLP	**RBF**	**ELM**	**ESN**	**LSTM**
RMSE	17571.05	20020.14	7155.19	**4568.06**	20685.19
MSE	308741639.30	400806089.20	51196802.39	**20867176.49**	427877224.99
MAE	15765.70	18407.43	6646.02	**3534.57**	19095.47
MAPE	0,51%	0,62%	0,22%	**0,12%**	0,62%

Fonte: autoria própria (2025)

O gráfico 4 mostra o *boxplot* do MSE das redes para a série do PR. É possível perceber que a MLP teve a menor dispersão, o que significa que essa é a rede com a melhor previsibilidade.

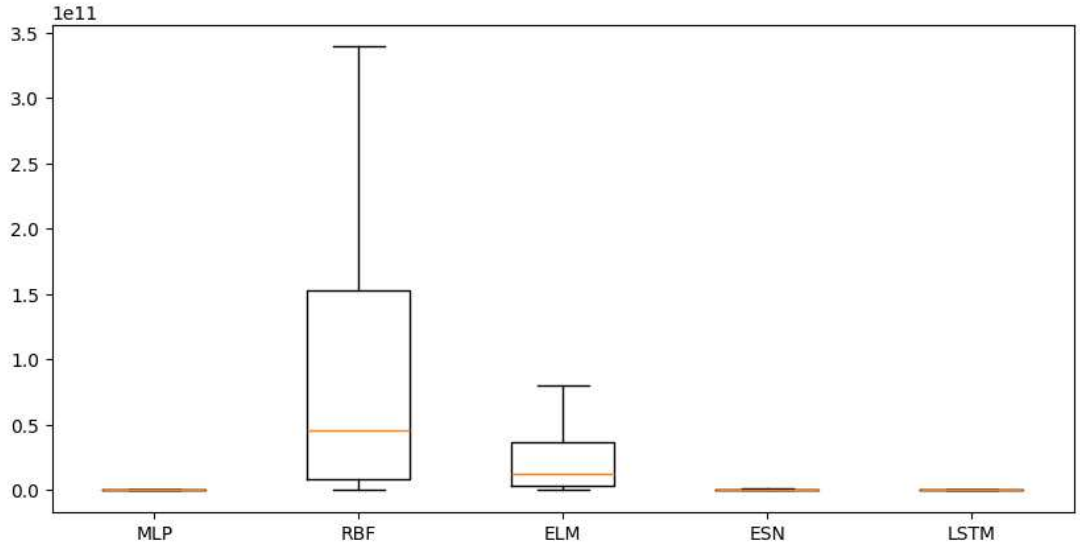
O gráfico 5 mostra a curva prevista pela ESN, bem como a curva de dados reais, para que seja possível uma análise visual dos dados gerados e o efeito do erro de previsão neles.

4.2.2 Santa Catarina

Para o estado de Santa Catarina, a rede com o melhor desempenho foi a ELM, tendo valores de RMSE, MSE, MAE e MAPE melhores que todas as outras.

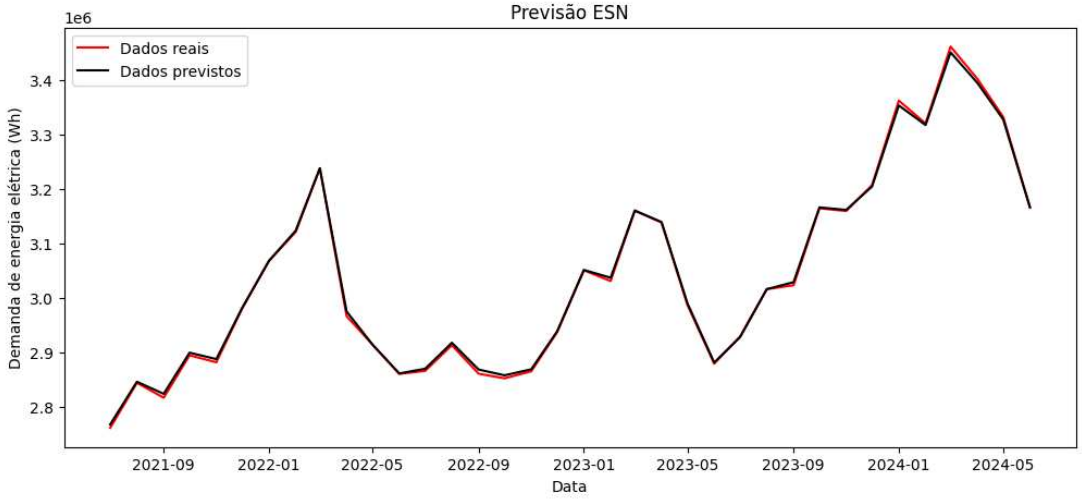
O *boxplot* do MSE das redes para a série de SC é exibido no gráfico 6. Mais uma vez, a MLP teve a menor dispersão, confirmando, assim, sua melhor previsibilidade.

Gráfico 4 – Boxplot dos MSEs para a série do PR



Fonte: autoria própria (2025)

Gráfico 5 – Previsão da ESN - PR



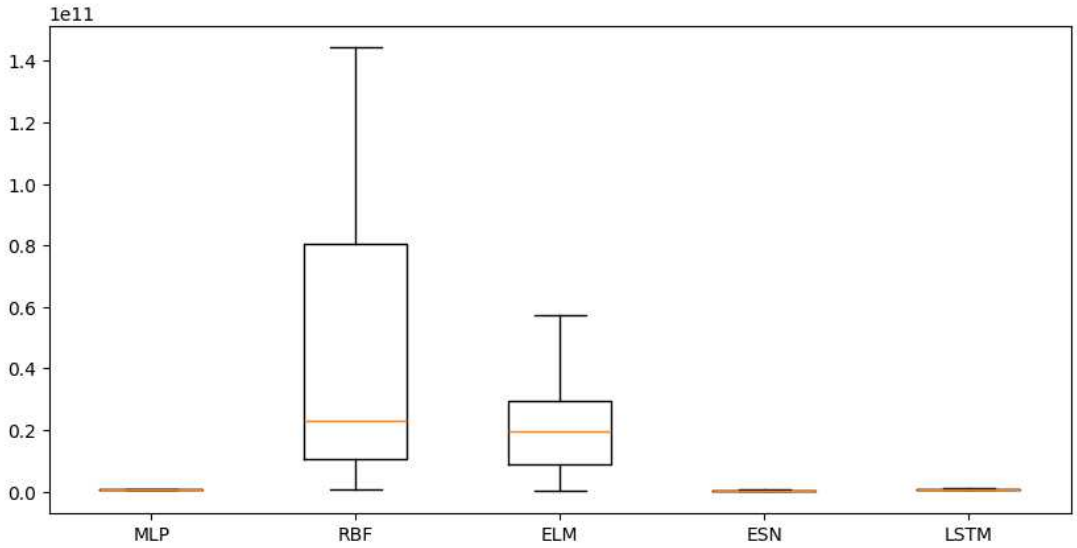
Fonte: autoria própria (2025)

Tabela 7 – Erros - Santa Catarina

	MLP	RBF	ELM	ESN	LSTM
RMSE	22448.55	23607.12	**6934.77**	7031.22	23244.47
MSE	503937358.00	557296004.91	**48091003.07**	49438109.99	540305326.62
MAE	20331.04	22735.82	**6202.88**	5804.27	21183.43
MAPE	0,81%	0,95%	**0,25%**	0,24%	0,85%

Fonte: autoria própria (2025) (2025)

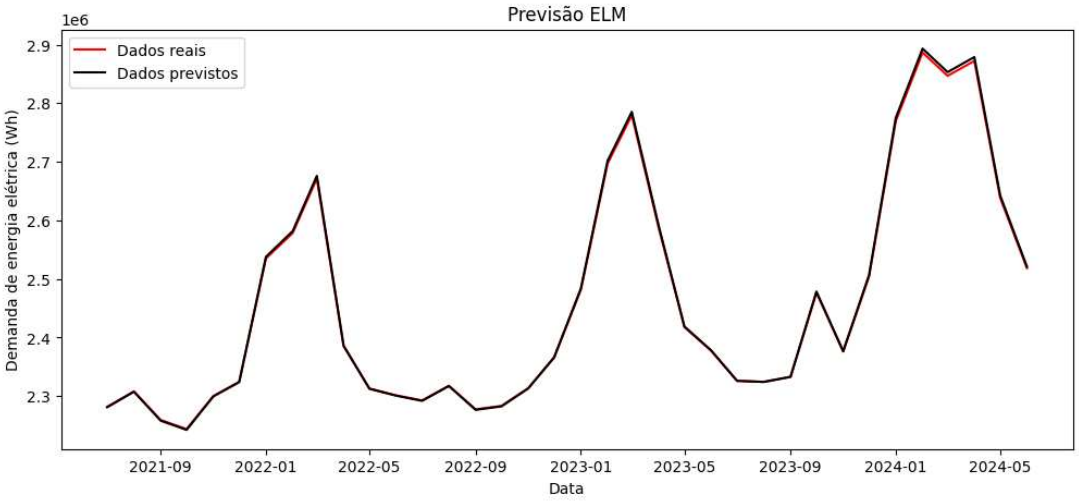
Gráfico 6 – Boxplot dos MSEs para a série do SC



Fonte: autoria própria (2025)

O gráfico 7 mostra a curva prevista pela ELM juntamente à curva de dados reais, a fim de possibilitar uma análise visual dos dados obtidos e o efeito do erro de previsão neles.

Gráfico 7 – Previsão da ELM - SC



Fonte: autoria própria (2025)

4.2.3 Rio Grande do Sul

A rede com o melhor desempenho para o estado do Rio Grande do Sul também foi a ESN. É possível notar que esta série teve uma média de valores de erro maior do que as outras.

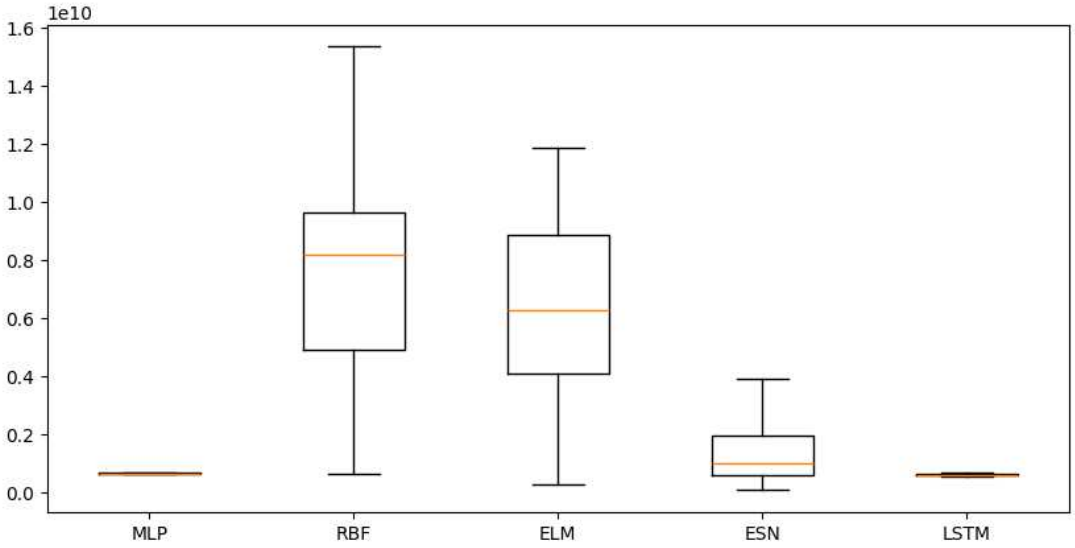
Tabela 8 – Erros - Rio Grande do Sul

	MLP	RBF	ELM	ESN	LSTM
RMSE	25300.71	25574.60	16794.90	**8147.72**	23007.69
MSE	640125802.52	654060048.23	282068662.87	**66385637.99**	529353715.37
MAE	18387.19	18062.64	15823.42	**7178.86**	16509.58
MAE	0,67%	0,66%	0,62%	**0,28%**	0,60%

Fonte: autoria própria (2025)

O gráfico 8 contém o *boxplot* do MSE das redes para a série do RS. É perceptível a menor dispersão da MLP em comparação com as outras redes, confirmando, assim, sua melhor previsibilidade.

Gráfico 8 – Boxplot dos MSEs para a série do RS



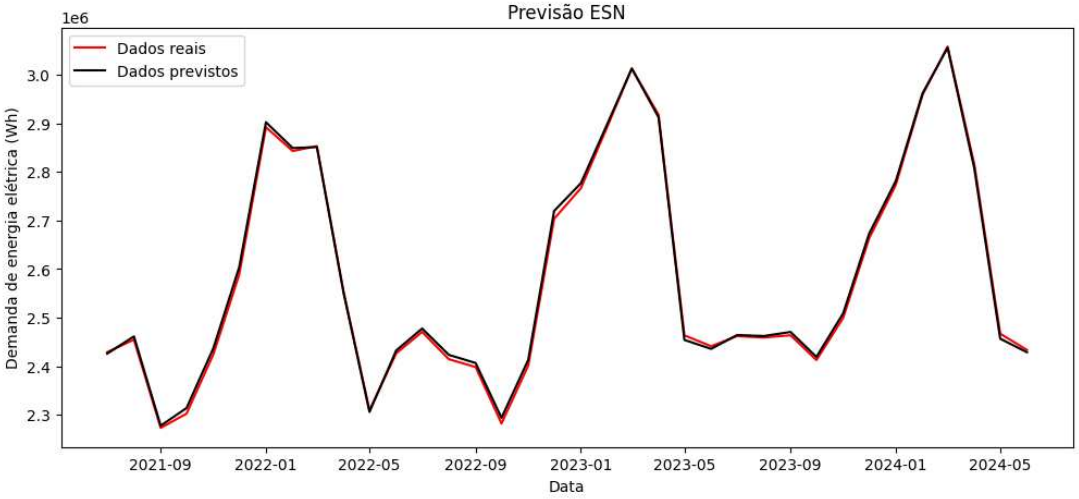
Fonte: autoria própria (2025)

O gráfico 9 mostra a curva de previsão obtida da ESN, assim como a curva de dados reais, para que seja possível uma análise visual dos dados gerados e sua diferença com a realidade.

4.3 Considerações finais

É possível perceber que a série do RS teve os maiores erros, e sua curva de previsão gerada pela ESN é a que mais destoa da curva dos dados reais. Mas, via de regra, as redes desenvolvidas obtiveram previsões satisfatórias para as três séries temporais, sendo a MLP a rede com a maior capacidade de generalização nos três casos.

Gráfico 9 – Previsão da ESN - RS



Fonte: autoria própria (2025)

5 CONCLUSÕES

Este trabalho teve como objetivo realizar um estudo sobre a importância de um fornecimento de energia elétrica seguro e implementar a previsão de séries temporais utilizando Redes Neurais Artificiais (RNA) com o objetivo de prever a demanda de energia elétrica, a fim de auxiliar no planejamento para seu fornecimento.

Foram estudadas as RNAs do tipo Perceptron de Múltiplas Camadas (MLP), Funções de Base Radial (RBF), Máquina de Aprendizado Extremo (ELM), Máquina de Estado de Eco (ESN) e Memória de Longo-Curto Prazo (LSTM) para a previsão de séries temporais de dados de demanda de energia elétrica nos estados do Paraná (PR), Santa Catarina (SC) e Rio Grande do Sul (RS).

Referente aos resultados das redes, observou-se uma grande assertividade vinda dos modelos, sendo a ESN e a ELM as redes com melhores resultados e a RBF a rede com os piores. Porém, a rede com a menor dispersão e, consequentemente, a melhor previsibilidade, foi a MLP. Isso evidencia que as RNAs desenvolvidas podem realizar previsões adequadas de demanda de energia elétrica. Assim, pode-se assegurar que as RNAs desenvolvidas e analisadas, principalmente a MLP, são viáveis dentro do propósito deste estudo, conseguindo chegar ao objetivo declarado no início do trabalho.

Para trabalhos futuros, é possível complementar as bases de dados utilizadas com mais amostras, para que o treinamento seja mais adequado e as redes obtenham uma maior capacidade de generalização. Também é possível aplicar as redes desenvolvidas em outras séries temporais de demanda de energia elétrica, como, por exemplo, séries de outros estados, ou até mesmo dos estados utilizados neste trabalho, mas com uma granularidade diferente, como dados diários ou horários.

REFERÊNCIAS

COIMBRA, Tainá de Souza. **Métodos de Pré-processamento no Uso de Modelos Lineares e de Redes Neurais para Previsão de Demanda de Energia**. 2023. 66 f. Dissertação (Mestrado em Engenharia Elétrica) – Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, SP, dez. 2023. Disponível em: https://www.repositorio.unicamp.br/acervo/detalhe/1384041. Acesso em: 5 dez. 2023.

ENERGÉTICA, Empresa de Pesquisa. **Consumo Mensal de Energia Elétrica por Classe (regiões e subsistemas)**. [*S. l.: s. n.*], ago. 2024. The GNOME Project. Disponível em: https://www.epe.gov.br/pt/publicacoes-dados-abertos/publicacoes/consumo-de-energia-eletrica. Acesso em: 5 dez. 2023.

FU, Rui; ZHANG, Zuo; LI, Li. Using LSTM and GRU neural network methods for traffic flow prediction*. In*: 2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC). [*S. l.: s. n.*], 2016. p. 324–328. DOI: 10.1109/YAC.2016.7804912.

GÉRON, Aurélien. **Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow**. Sebastopol, CA, EUA: O'Reilly Media, set. 2019. 272 p. ISBN 9781492032649.

HARRISON, Matt. **Machine Learning – Guia de Referência Rápida**. São Paulo, SP, BR: Novatec Editora, dez. 2019. 272 p. ISBN 9788575228180.

HAYKIN, Simon. **Redes Neurais: Princípios e prática**. São Paulo, SP, BR: Artmed Editora, dez. 2008. 903 p. ISBN 0132733501.

HOCEVAR, Luciano Sergio; ALVES, Carine Tondo; PEREIRA, Jadiel dos Santos. Matriz elétrica e insegurança energética*. In*: 77ª SEMANA OFICIAL DA ENGENHARIA E DA AGRONOMIA (SOEA), out. 2022, Goiânia, GO. **Proceedings [...].** [*S. l.: s. n.*], 2022. Congresso Técnico-Científico da Engenharia e da Agronomia (CONTECC).

HOCHREITER, Sepp; SCHMIDHUBER, Jürgen. Long Short-Term Memory. **Neural Comput.**, MIT Press, Cambridge, MA, USA, v. 9, n. 8, p. 1735–1780, nov. 1997. ISSN 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. Disponível em: https://doi.org/10.1162/neco.1997.9.8.1735.

HUANG, Guang-Bin; ZHU, Qin-Yu; SIEW, Chee-Kheong. Extreme learning machine: a new learning scheme of feedforward neural networks*. In*: 2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541). [*S. l.: s. n.*], 2004. v. 2, 985–990 vol.2. DOI: 10.1109/IJCNN.2004.1380068.

JAEGER, Herbert. The"echo state"approach to analysing and training recurrent neural networks-with an erratum note'. **Bonn, Germany: German National Research Center for Information Technology GMD Technical Report**, v. 148, jan. 2001.

LECUN, Yann; BENGIO, Y.; HINTON, Geoffrey. Deep Learning. **Nature**, v. 521, p. 436–44, mai. 2015. DOI: 10.1038/nature14539.

MORETTIN, Pedro Alberto; TOLOI, Clélia Maria de Castro. **Análise de Séries Temporais**. São Paulo, SP, BR: Editora Blucher, jan. 2006. 564 p. ISBN 9788521203896.

OZTURK, Mustafa C.; XU, Dongming; PRÍNCIPE, José C. Analysis and Design of Echo State Networks. **Neural Computation**, v. 19, n. 1, p. 111–138, jan. 2007. ISSN 0899-7667. DOI: 10.1162/neco.2007.19.1.111. eprint: https://direct.mit.edu/neco/article-pdf/19/1/111/816757/neco.2007.19.1.111.pdf. Disponível em: https://doi.org/10.1162/neco.2007.19.1.111.

RIVERO, José Ricardo Magalhães; JUNIOR, Cleber Almeida Corrêa; CORRÊA, Rosilene Abreu Portella. Application of artificial neural networks to predict the behavior of stocks. *In*: 3. INTERNATIONAL Journal of Advanced Engineering Research and Science. [*S. l.: s. n.*], 2023. v. 10, p. 1–6. DOI: 10.22161/ijaers.103.1.

SILVA, Ivan Nunes da *et al.* Artificial Neural Network Architectures and Training Processes. *In*: ARTIFICIAL Neural Networks : A Practical Course. Cham: Springer International Publishing, 2017. p. 21–28. ISBN 978-3-319-43162-8. DOI: 10.1007/978-3-319-43162-8_2. Disponível em: https://doi.org/10.1007/978-3-319-43162-8_2.

SILVA, Ivan Nunes da; SPATTI, Danilo Hernane; FLAUZINO, Rogério Andrade. **Redes Neurais Artificiais para engenharia e ciências aplicadas - curso prático**. São Paulo, SP, BR: Artliber Editora, dez. 2010. 399 p. ISBN 9788588098534.

SIQUEIRA, Hugo Valadares. **Máquinas Desorganizadas para Previsão de Séries de Vazões**. 2013. 218 f. Tese (Doutorado em Engenharia Elétrica) – Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, Campinas, SP, nov. 2013. Disponível em: http://paginapessoal.utfpr.edu.br/hugosiqueira/tese-e-dissetacao/Tese%20-%20Hugo%20Valadares%20Siqueira.pdf/view. Acesso em: 5 dez. 2023.

TIDRE, PVV; BIASE, NGG; SILVA, MI de S. Utilização dos modelos de séries temporais na previsão do consumo mensal de energia elétrica da região norte do Brasil. **Matemática e Estatística em Foco**, v. 1, n. 1, p. 57–66, 2013.

VALOIS COELHO, Ilsa Maria; CARTAXO, Elizabeth Ferreira. Universalização da energia elétrica: uma análise política da distribuição de energia e da sua importância sócio-ambiental para o Amazonas. **Procedings of the 5th Encontro de Energia no Meio Rural**, SciELO Brasil, 2004.