Predição da Velocidade do Vento Utilizando Redes Neurais *Wavelet*

Áldri de Souza Magalhães * Thiago Pereira das Chagas **

* Programa de Pós-Graduação em Mecatrônica, Universidade Federal da Bahia, BA, (e-mail: aldrimagalhaes@gmail.com).

** Laboratório de Mecatrônica, Universidade Estadual de Santa Cruz, BA (e-mail: tpchagas@uesc.br)

Abstract: In this work the wind speed is predicted for wind power application estimation. Data obtained from an anemometric station were used for the application of two artificial intelligence techniques, the artificial neural network and the wavelet neural network. The autocorrelation function was used to determine the amount of time-delayed inputs for the time series prediction, which allowed the use of data in order to increase the extrapolation capacity of the networks. The comparison of the predictions obtained from the methods was performed through the mean absolute percentage error, as well as the coefficient of determination, which proved a better generalization capacity and precision by the wavelet neural network.

Resumo: Neste trabalho é feita a predição da velocidade do vento para aplicação na estimação de potência gerada por aerogeradores. Foram utilizados dados obtidos de uma estação anemométrica para a aplicação de duas técnicas de inteligência artificial, a rede neural artificial e a rede neural *wavelet*. A função de autocorrelação foi empregada na determinação da quantidade de entradas atrasadas no tempo para a predição da série temporal, o que possibilitou a utilização dos dados de forma a aumentar a capacidade de extrapolação das redes. A comparação das predições obtidas a partir dos métodos foi realizada por meio do erro percentual absoluto médio, assim como do coeficiente de determinação, o que comprovou uma melhor capacidade de generalização e precisão por parte da rede neural *wavelet*.

Keywords: Artificial Neural Network, Wavelet Neural Network, Wind Speed Prediction.

Palavras-chaves: Rede Neural Artificial; Rede Neural *Wavelet*; Previsão da Velocidade do Vento.

1. INTRODUÇÃO

De acordo com ABEEólica (2019b), foram instalados aproximadamente 50 GW de energia eólica no mundo em 2017. No final de 2018, o Brasil chegou à marca de 14,34 GW em capacidade instalada de energia eólica, que equivale à usina de Itaipu, a maior do país (ABEEólica, 2019a). Apesar da capacidade de geração instalada no país, a conexão com uma rede de transmissão e distribuição de energia elétrica tem sua confiabilidade influenciada pelo caráter descontínuo da velocidade dos ventos.

Reis (2011) descreve o complexo eólico composto por aerogeradores, informando que a potência do vento é diretamente proporcional ao cubo de sua velocidade. Portanto, pequenas oscilações na comportamento do vento podem ocasionar grandes variações de potência. Com a previsão da velocidade do vento a partir de modelos que utilizam dados previamente coletados, pode-se prever a potência gerada (Doucoure et al., 2016; Senkal e Ozgonenel, 2013).

Algumas das ferramentas convencionais empregadas na predição consistem em modelos estatísticos, como o ARMA (*autoregressive moving average*) e ARIMA (*autoregressive integrated moving average*). Entretanto, técnicas de inteligência artificial, como, por exemplo, K vizinhos

mais próximos (KNN - *k-nearest neighbors*) e rede neural artificial (RNA), têm sido utilizadas na predição de séries temporais (Ferraz et al., 2017). As RNAs trabalham com não linearidades presentes nos sistemas reais e apresentam resultados promissores (Chen e Folly, 2018). Contudo, cada técnica de predição possui sua desvantagem, por isso modelos híbridos do ARIMA com a RNA, por exemplo, têm sido estudados (Nair et al., 2017; Chang et al., 2016).

Para melhorar o desempenho de predição por rede neural isolada ou em conjunto com um modelo estatístico, é interessante aperfeiçoar o desempenho desta. Dewangan et al. (2017) comprovam que uma forma de melhoria é o emprego da rede neural *wavelet* (RNW), que une as vantagens da RNA e decomposição *wavelet*.

Na predição de séries temporais com modelos estatísticos é necessário selecionar as amostras atrasadas no tempo, usualmente determinadas através da análise do correlograma, para determinar as entradas do modelo (Nunnari et al., 2014). Neste trabalho propõe-se como modelo uma RNA, criada e treinada utilizando o *neural network toolbox* do MATLAB, e uma RNW implementada aqui, em MATLAB, a partir da teoria apresentada por Alexandridis e Zapranis (2014). São variadas a forma como os dados atrasados são aplicados a partir da análise do correlograma

da série temporal. Como resultado, esse estudo apontou numa melhoria na capacidade de generalização das redes neurais, além de confirmar o melhor desempenho da RNW.

Este trabalho é dividido em oito seções, sendo que a seção 2 é definida a função de autocorrelação e sua aplicação. As seções 3 e 4, introduzem a RNA e aprofundam a teoria envolvendo a RNW. Para finalizar a fundamentação teórica, são mostrados e explicados na seção 5 os parâmetros utilizados para avaliar os modelos neurais. Por fim, nas seções 6, 7 e 8, são apresentadas a metodologia com os resultados obtidos sendo discutidos e a conclusão deste trabalho.

2. FUNÇÃO DE AUTOCORRELAÇÃO

A autocorrelação r , definida por (1), avalia o quanto uma amostra p de um determinada série temporal $Z = Z_1, Z_2, \dots, Z_p, \dots, Z_n$ espaçada igualmente no tempo com n amostras está relacionada aos demais valores do mesmo sinal. A série temporal com média $\bar{Z}$ é defasada em relação a si mesma para realizar o cálculo da correlação entre os valores da série original e a defasada de d_e amostras, obtendo-se o correlograma que consiste em um gráfico que expressa a autocorrelação em relação aos atrasos (Box et al., 1976; Hamilton, 1994).

$$r(d_e) = \frac{\sum_{p=1}^{n-d_e} (Z_p - \bar{Z})(Z_{p+d_e} - \bar{Z})}{\sum_{p=1}^n (Z_p - \bar{Z})^2} \quad (1)$$

Portanto, do correlograma, pode-se selecionar as amostras que apresentam índices consideráveis de autocorrelação, para comporem as entradas atrasadas que serão informadas ao modelo de previsão utilizado, como feito por Nunnari et al. (2014).

3. REDE NEURAL ARTIFICIAL

Haykin (2001) conceitua a RNA como um modelo que visa mimetizar computacionalmente as funções do cérebro humano através de estruturas constituídas por neurônios artificiais. A capacidade de aprendizagem da RNA é obtido com ajuste de seus parâmetros através das técnicas de otimização, que aproxima o comportamento de um sistema sem conhecer o seu processo interno. Os dados são processados por neurônios constituídos de sinapses que conectam as entradas x ao neurônio e são caracterizadas por pesos ou parâmetros W . Devido a esta característica, os pesos determinam a relevância que o sinal de entrada tem para a saída do neurônio.

Há diversas formas dos neurônios serem estruturados, no caso da rede alimentada adiante, são agrupados em camadas de entrada, oculta (podendo haver mais de uma) e de saída.

A previsão de séries temporais se baseia em utilizar dados com atrasos no tempo para a sua execução. Portanto, a camada de entrada é composta pelo dado atual e os valores atrasados da série temporal (Ferraz et al., 2017).

4. REDE NEURAL WAVELET

Uma *wavelet* ψ é definida como uma função com base de suporte compacto, que é uma propriedade na qual sua

imagem é diferente de zero em uma extensão finita e igual a zero em todo o restante de seu domínio em um problema bidimensional (Bianchi, 2006). Alexandridis e Zaprani (2014) definem uma família *wavelet* como um conjunto de funções geradas pela dilatação e translação de uma *wavelet* ψ (*wavelet* mãe), que deve satisfazer a integral da função ser nula.

As *wavelets* filhas são definidas por

$$\psi_{a,b}(x) = \frac{1}{\sqrt{a}} \psi\left(\frac{x-b}{a}\right), \quad (2)$$

em que a é o parâmetro de dilatação, ou de escala, b é o parâmetro de translação, ou de deslocamento. O termo $1/\sqrt{a}$ é usado para normalizar $\|\psi_{a,b}(x)\| = 1$. Como a *wavelet* é uma função com domínio limitado, ela pode ser muito local se seu parâmetro de dilatação for muito pequeno, ou pode ficar fora do domínio de interesse, se o parâmetro de translação não for escolhido apropriadamente (Oussar e Dreyfus, 2000).

Alguns exemplos de *wavelets* ψ são *gaussian derivative*, *mexican hat* e *morlet wavelet* (Alexandridis e Zaprani, 2014).

4.1 Estrutura da rede neural wavelet

A RNW é uma rede neural com função de base radial, que é uma função sobre números reais cujos valores dependem apenas da distância a partir da origem, tal que $f(v) = f(\|v\|)$. Alexandridis e Zaprani (2014) propõem a estrutura da RNW aplicando a *wavelet* multidimensional na camada oculta e com uma única saída alimentada adiante. Neste trabalho, é utilizada a estrutura da RNW para o caso de múltiplas saídas, conforme a Figura 1.

Figura 1. Rede *wavelet* alimentada adiante.

No modelo de neurônio empregado na camada oculta da RNW, Figura 2, percebe-se que cada sinal de entrada $x_{(p,i)}$, em que $i = 1, 2, \dots, m$ e m é o número de entradas, tem-se uma *wavelet* transladada e dilatada, o que pode ser comparado ao processo realizado pela transformada *wavelet*.

Figura 2. Modelo de um neurônio da camada oculta de uma RNW.

A entrada da *wavelet* filha $z_{(i,g,p)}$, definida por (2), consiste em um conjunto de dados $x_{(p,i)}$ ponderado pelos parâmetros de dilatação $a_{(i,g)}$ e translação $b_{(i,g)}$, que conectam a entrada i ao neurônio da camada oculta $g = 1, 2, \dots, l$, com l é o número de neurônios na camada oculta,

$$z_{(i,g,p)} = \frac{x_{(p,i)} - b_{(i,g)}}{a_{(i,g)}}.$$

Neste trabalho é aplicado como *wavelet* mãe $\psi_{(i,g,p)}$ a *mexican hat*,

$$\psi_{(i,j,p)} = \left(1 - z_{(i,g,p)}^2\right) e^{-\frac{1}{2}z_{(i,g,p)}^2}.$$

Como há múltiplas entradas, utiliza-se a *wavelet* multi-dimensional $\Psi_{(p,g)}$, proposta por Zhao et al. (1998), que corresponde à saída de cada neurônio da camada oculta mostrado na Figura 2,

$$\Psi_{(p,g)} = \prod_{i=1}^m \psi_{(i,g,p)}.$$

Os neurônios na camada de saída possuem estrutura igual aos da RNA, conforme foi discutido na seção 3 e é mostrado na Figura 3. Portanto, matematicamente, as saídas da RNW, é

$$y_{(p,k)} = \zeta_s \left(\sum_{g=1}^l W_{s(k,g)} \Psi_{(p,g)} + \beta_{s(k)} \right), \quad (3)$$

em que $W_{s(k,g)}$ é o parâmetro que conecta o neurônio g da camada oculta com a saída k , com u sendo o número de saídas, $\beta_{s(k)}$ é o bias conectado à saída k , e ζ_s é a função de ativação aplicada aos neurônios da camada de saída, geralmente, é uma função linear.

4.2 Inicialização dos pesos da rede

Os parâmetros de translação e dilatação da função *wavelet*, que fazem parte dos pesos da RNW, devem ser inicializados de forma distinta em relação aos outros pesos. Zhang e Benveniste (1992) propuseram inicializar os parâmetros relacionados a *wavelet* com:

$$\begin{aligned} b_{(i,g)} &= 0,5(M_{(i)} + N_{(i)}) \\ a_{(i,g)} &= 0,2(M_{(i)} - N_{(i)}) \end{aligned}$$

Figura 3. Modelo não linear de um neurônio.

$M_{(i)}$ e $N_{(i)}$ são os valores máximos de cada entrada i do conjunto de dados x . Os demais pesos conectados aos neurônios na camada de saída são inicializados com valores aleatórios como na RNA.

O método proposto por Zhang e Benveniste (1992) é de fácil implementação e possui custo computacional insignificante, entretanto, não realiza uma busca pelo mínimo global da função de custo, que será abordada na próxima seção, e pode levar a um período extenso de otimização. Outros métodos podem ser utilizados na inicialização, como, por exemplo, *stepwise selection by orthogonalization* (Alexandridis e Zaprani, 2014), e a otimização por enxame de partículas (Parsopoulos e Vrahatis, 2010), podem ser empregados, respectivamente.

4.3 Treinamento com descida de gradiente

Depois da inicialização, os pesos são treinados com o intuito de minimizar uma função de custo, que varia para cada problema. No caso do ajuste do modelo por redes neurais, o objetivo consiste em calcular o erro cometido pela rede neural. Há diversas funções de custo, mas, geralmente, consistem da média quadrática do erro $E_{(p,k)}$ entre a saída k da rede $y_{(p,k)}$, definida por (3) no caso da RNW, e a saída k desejada correspondente $y_{d(p,k)}$, dado por

$$E_{(p,k)} = y_{d(p,k)} - y_{(p,k)}.$$

É proposto neste trabalho uma função custo L da média quadrática do erro considerando várias saídas da rede neural referente a iteração de treinamento i_t ,

$$L^{(i_t)} = \frac{1}{2 \cdot n \cdot u} \sum_{p=1}^n \sum_{k=1}^u E_{(p,k)}^2. \quad (4)$$

Levando em consideração (4) como a função de custo e o método da descida de gradiente, tem-se a lei de aprendizagem, utilizada por Alexandridis e Zaprani (2014):

$$w^{(i_t+1)} = w^{(i_t)} - \rho \nabla L[w]^{(i_t)} + \kappa(w^{(i_t)} - w^{(i_t-1)})$$

κ é o termo de *momentum speed* e ρ é a taxa de aprendizagem, ambos constantes visando aumentar a velocidade de treino e evitar oscilações. ρ e κ assumem um valor entre 0 e 1, mas geralmente, usa-se o intervalo entre 0 e 0.4, dependendo da aplicação e das amostras de treinamento (Alexandridis e Zaprani, 2014). Além disso, $\nabla L[w]^{(i_t)}$ é o

gradiente da função de custo em relação ao vetor de pesos $w = [W_o, \beta_o, W_s, \beta_s]$. Os gradientes são definidos como:

$$\nabla L[\beta_{s(k)}]^{(i_t)} = -\frac{1}{n \cdot u} \sum_{p=1}^n E_{(p,k)} \cdot \zeta'_{s(p,k)}$$

$$\nabla L[W_{s(k,g)}]^{(i_t)} = -\frac{1}{n \cdot u} \sum_{p=1}^n E_{(p,k)} \cdot \zeta'_{s(p,k)} \cdot \Psi_{(p,g)}$$

$$\nabla L[b_{(i,g)}]^{(i_t)} = \frac{1}{n \cdot u} \sum_{p=1}^n \sum_{k=1}^f E_{(p,k)} \cdot \zeta'_{s(p,k)} \cdot \frac{W_{s(k,g)}}{a_{(i,g)}} \cdot \psi_{(1,g,p)} \cdot \dots \cdot \psi'_{(i,g,p)} \cdot \dots \cdot \psi_{(m,g,p)}$$

$$\nabla L[a_{(i,g)}]^{(i_t)} = \frac{1}{n \cdot u} \sum_{p=1}^n \sum_{k=1}^f E_{(p,k)} \cdot \zeta'_{s(p,k)} \cdot z_{(i,g,p)} \cdot \frac{W_{s(k,g)}}{a_{(i,g)}} \cdot \psi_{(1,g,p)} \cdot \dots \cdot \psi'_{(i,g,p)} \cdot \dots \cdot \psi_{(m,g,p)}$$

A otimização da rede é um processo iterativo, e como tal deve-se definir critérios para a finalização da atualização. O ideal é que garantam a convergência da algoritmo para o mínimo global da função custo. Contudo, Haykin (2001) afirma que não há como se demonstrar que o algoritmo de retropropagação convergiu e não existem critérios bem definidos para encerrar sua operação. Além do mais, os tempos de aprendizagem podem ser longos. Aqui utilizam-se critérios de parada como na RNA convencional.

5. PARÂMETROS DE AVALIAÇÃO DO PREDITOR

Para medir o desempenho dos modelos de predição, utilizam-se funções baseadas no erro, (4.3). Uma é o coeficiente de determinação R^2 ,

$$R^2_{(k)} = 1 - \frac{\sum_{p=1}^n E_{(p,k)}^2}{\sum_{p=1}^n (y_{d(p,k)} - \bar{y}_d)^2}, \quad (5)$$

que assume valores de interesse no intervalo de 0 a 1, sendo que quanto mais perto do valor unitário, maior a capacidade da rede de generalizar. Valores fora desse intervalo, indicam que o treinamento não foi satisfatório.

Em (5), o erro E é ponderado pela dispersão do conjunto de dados y_d em relação a sua própria média $\bar{y}_d$. O erro é mais significativo em regiões com valores próximos da média dos dados, portanto um R^2 perto do valor unitário indica que o modelo apresenta uma maior acurácia. Estes valores são mais significativos para predição de vento com aplicação à geração eólica, pois, os aerogeradores utilizam controladores de forma a girarem em velocidade constante.

Outro fator levado em consideração é o erro MAPE (*Mean Absolute Percentage Error*),

$$MAPE_{(k)} = \frac{1}{n} \sum_{p=1}^n \left| \frac{E_{(p,k)}}{y_{(p,k)}} \right| \cdot 100\%, \quad (6)$$

que expressa a amplitude média da magnitude do erro.

6. METODOLOGIA

No MATLAB há o *Neural Network Toolbox* para a simulação e treinamento de uma RNA. Foram utilizados os valores padrões das variáveis do *toolbox*. Além disso, para o treinamento, aplicou-se o método da descida de gradiente, como função de custo, utilizou-se o MSE (*mean square error*). Em relação a função de ativação, adotou-se a tangente hiperbólica na camada oculta e a linear na saída.

Ao contrário da RNA, a RNW não está implementada no MATLAB, Scilab, entre outros, por isso foi implementada por linhas de código em MATLAB a partir das equações presentes na seção 4. A *wavelet* selecionada para ser empregada na camada oculta da RNW foi a *mexican hat* e a função de ativação na saída foi a linear. Com exceção da forma de implementação da rede e de seu treinamento, utilizou-se os mesmos métodos para ambas as redes. Além de aplicar uma taxa de aprendizagem $\rho = 0,4$ e como critérios de parada o valor dos gradientes e a variação do RMSE (*root mean square error*) como menores do que 10^{-5} .

7. RESULTADOS E DISCUSSÕES

Os dados utilizados para a simulação são os mesmos empregados por Ferraz et al. (2017), sendo trinta minutos sua taxa de amostragem.

Aplicando (1) aos dados, foi obtido o correlograma mostrado na Figura 4, há valores de autocorrelação próximos a zero, indicando uma componente aleatória na série, além da presença de picos apontar para a sazonalidade, enquanto que o comportamento decrescente ou crescente dos valores de autocorrelação é um fator de tendência.

Figura 4. Correlograma dos dados destinados ao treinamento.

Considera-se que uma autocorrelação superior a 0,5 é próxima suficiente da correlação perfeita (valor unitário) (Ferraz et al., 2017). Por tanto, além da última amostra disponível (zero atraso), as amostras correspondentes aos atrasos 1-6 e 41-51 são as entradas atrasadas que serão informadas a rede neural, isso era esperado pela componente sazonal diária presente no conjunto de dados. Por

exemplo, se a última amostra disponível foi às 12 horas, as amostras anteriores até 9 horas assim como as de 10:30 à 15:30 horas do dia anterior serão informadas para fazer a predição até as 18 horas.

Apesar de determinar a quantidade de entradas atrasadas dessa forma, foram exploradas outras possibilidades, como utilizar apenas 1-6 (7 entradas) ou 1-13 atrasos (14 entradas), que corresponderia à mesma quantidade de entradas atrasadas que 1-6 mais 45-51 (14 entradas).

Além de realizar o teste comparando as topologias de entradas atrasadas, a quantidade de neurônio na camada oculta foi determinada via média do coeficiente de determinação, (5), das saídas k da rede neural, acrescentando cinco neurônios começando inicialmente com 5 e variando até 50 neurônios. Na Figura 5, os resultados desses testes com a RNA mostram que o aumento de entradas proporciona, em média, uma maior capacidade de generalização, sendo que a estrutura com 1-6 e 45-51 atrasos foi a que apresentou os valores mais próximos do unitário.

Figura 5. Comportamento do R^2 com a variação de neurônios e entradas da RNA.

Esse mesmo comportamento foi observado para a RNW na Figura 6. Contudo, com a variação dos neurônios o melhor índice para a RNA e RNW foi com 10 e 5 neurônios, respectivamente. No caso da RNA a estrutura com 5 neurônios apresenta um resultado próximo e, na prática, seria preferível selecionar esse modelo para a simulação por ter menos parâmetros. Mas neste trabalho têm-se o objetivo de comparar os melhores resultados, assim utiliza-se 1-6 e 45-51 atrasos para ambas as redes e 10 e 5 neurônios para a RNA e RNW, respectivamente.

Avaliou-se o erro MAPE, definido por (6), referente a cada uma das saídas da rede como mostrado na Figura 7. Com o avanço nos passos de predição o resultado se degrada, sendo que a RNW apresenta os menores erros em comparação com a RNA. Predições após 2 horas a RNA resulta em erros MAPE próximos à RNW com 1 hora a mais de predição. Ressalta-se que as redes neurais são mais indicadas a predições longas em comparação com modelos estatísticos (Chen e Folly, 2018; Ferraz et al., 2017).

Figura 6. Comportamento do R^2 com a variação de neurônios e entradas da RNW.

Figura 7. Comportamento do MAPE no decorrer do horizonte de predição.

A predição para um passo à frente para o período de uma semana (336 amostras) feita com os melhores modelos da RNA e da RNW é mostrada na Figura 8. Quanto maior a distância na predição maiores as diferenças entra as curvas reais e preditas, como acontece na predição para seis passos à frente, Figura 9.

8. CONCLUSÃO

Neste trabalho foi proposto uma metodologia para determinar as amostras anteriores que serão definidas como entradas do modelo neural, conforme ideia proposta por Nunnari et al. (2014), comparando a RNA com a RNW. Esta última apresenta melhor desempenho em problemas de predição do que RNA, por unir as vantagens proporcionadas pela decomposição *wavelet*. Para mostrar a eficácia desse método, foram realizados testes modificando a quantidade de entradas a serem informadas ao modelo e variando a quantidade de neurônios. O R^2 possibilitou

Figura 8. Predição de um passo a frente para a RNA e RNW.

Figura 9. Predição de seis passos a frente para a RNA e RNW.

escolher a topologia de entradas atrasadas assim como a quantidade de neurônios na camada oculta de forma que a RNA e RNW tivessem a maior capacidade de extrapolação, sendo que esta última foi superior neste aspecto. Avaliando esses modelos com o MAPE, os erros aumentaram à medida que aumentou-se os passos de predição, mas a RNW apresentou os menores erros em todo o horizonte de predição.

O tempo de treinamento da RNW foi superior, pois depende fortemente da inicialização dos seus pesos na camada oculta, contudo, como é feito offline, não consiste em um problema. Para aplicação, pode-se empregar outros métodos de inicialização ou, então, técnicas heurísticas como algoritmos de treinamento. Por esses motivos, a comparação em relação ao tempo de treinamento não foi discutida neste trabalho, focando no desempenho da rede treinada.

REFERÊNCIAS

ABEEólica (2019a). Energia eólica ultrapassa marca de 14 gw de capacidade instalada. URL <http://abeeolica.org.br/noticias/energia-eolica-ultrapassa-marca-de-14-gw-de-capacidade-instalada/>.
ABEEólica (2019b). Mais de 50 gw de energia eólica foram instalados no mundo em 2017. URL <http://abeeolica.org.br/noticias/mais-de-50-gw-de-energia-eolica-foram-instalados-no-mundo-em-2017/>.
Alexandridis, A.K. e Zapanis, A.D. (2014). *Wavelet neural networks with applications in financial engineering, chaos, and classification*. Wiley.

Bianchi, M.F.d. (2006). *Extração de características de imagens de faces humanas através de wavelets, PCA e IMPCA*. Tese de mestrado, Escola de Engenharia de São Carlos da USP. doi:10.11606/D.18.2006.tde-10072006-002119.
Box, G.E.P., Jenkins, G.M., Reinsel, G.C., e Ljung, G.M. (1976). *Time series analysis: forecasting and control*. Holden-Day.
Chang, G.W., Lu, H.J., Hsu, L.Y., e Chen, Y.Y. (2016). A hybrid model for forecasting wind speed and wind power generation. In *2016 IEEE Power and Energy Society General Meeting (PESGM)*, 1–5. doi:10.1109/PESGM.2016.7742039.
Chen, Q. e Folly, K.A. (2018). Comparison of three methods for short-term wind power forecasting. In *2018 International Joint Conference on Neural Networks (IJCNN)*. doi:10.1109/IJCNN.2018.8489472.
Dewangan, C.L., Singh, S.N., e Chakrabarti, S. (2017). Solar irradiance forecasting using wavelet neural network. In *2017 IEEE PES Asia-Pacific Power and Energy Engineering Conference (APPEEC)*, 1–6. doi:10.1109/APPEEC.2017.8308987.
Doucoure, B., Agbossou, K., e Cardenas, A. (2016). Time series prediction using artificial wavelet neural network and multi-resolution analysis: application to wind speed data. *Renewable Energy*, 92(C), 202–211.
Ferraz, R.S.F., da C. Cruz, F., Ferraz, R.S.F., Correia, A.F.S., e de Simas, E.F. (2017). Previsão multi-passos da velocidade do vento através de redes neurais artificiais. *THE 12 th Latin-American Congress on Electricity Generation and Transmission - CLAGTEE 2017*.
Hamilton, J.D. (1994). *Time series analysis*. Princeton University Press, Princeton, NJ.
Haykin, S. (2001). *Redes neurais: princípios e prática*. Prentice Hall, Inc.
Nair, K.R., Vanitha, V., e Jisma, M. (2017). Forecasting of wind speed using ann, arima and hybrid models. In *2017 International Conference on Intelligent Computing, Instrumentation and Control Technologies (ICICICT)*, 170–175. doi:10.1109/ICICICT1.2017.8342555.
Nunnari, S., Fortuna, L., e Gallo, A. (2014). Wind time series modeling for power turbine forecasting. *International Journal of Electrical Energy*.
Oussar, Y. e Dreyfus, G. (2000). Initialization by selection for wavelet network training. *Neurocomputing*, 34, 131–143.
Parsopoulos, K.E. e Vrahatis, M.N. (2010). *Particle Swarm Optimization and Intelligence: Advances and Applications*. Information Science Reerence.
Reis, L.B.d. (2011). *Geração de energia elétrica*. Manole, 2 ed.
Senkal, S. e Ozgonenel, O. (2013). Performance analysis of artificial and wavelet neural networks for short term wind speed prediction. In *2013 8th International Conference on Electrical and Electronics Engineering (ELECO)*, 196–198. doi:10.1109/ELECO.2013.6713830.
Zhang, Q. e Benveniste, A. (1992). Wavelet networks. *IEEE Trans Neural Networks*, 3(6), 889–98. doi:10.1109/72.165591.
Zhao, J., Chen, B., e Shen, J. (1998). Multidimensional non-orthogonal wavelet sigmoid basis function neural network for dynamic process fault diagnosis. *Computers & Chemical Engineering*, 23(1), 83–92.