UNIVERSIDADE FEDERAL DO CEARÁ

CENTRO DE TECNOLOGIA

DEPARTAMENTO DE ENGENHARIA ELÉTRICA

CURSO DE GRADUAÇÃO EM ENGENHARIA ELÉTRICA

LUCAS TAVARES DA SILVA

APRENDIZADO DE MÁQUINA APLICADO NA PREVISÃO DA GERAÇÃO DE ENERGIA ELÉTRICA DE UMA USINA SOLAR FOTOVOLTAICA

FORTALEZA

2022

LUCAS TAVARES DA SILVA

APRENDIZADO DE MÁQUINA APLICADO NA PREVISÃO DA GERAÇÃO DE ENERGIA ELÉTRICA DE UMA USINA SOLAR FOTOVOLTAICA

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Elétrica do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia Elétrica.

Orientadora: Profª. PhD. Ruth Pastôra Saraiva Leão

Coorientador: Eng. André Wagner de Barros Silva

FORTALEZA

2022

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

S581a Silva, Lucas Tavares da.
Aprendizado de máquina aplicado na previsão da geração de energia elétrica de uma usina solar fotovoltaica / Lucas Tavares da Silva. – 2022.
58 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Centro de Tecnologia, Curso de Engenharia Elétrica, Fortaleza, 2022.
Orientação: Profa. Dra. Ruth Pastôra Saraiva Leão.
Coorientação: Prof. André Wagner de Barros Silva.

1. Aprendizado de Máquina. 2. Energia solar. 3. Inteligência Artificial. 4. Previsão. 5. Geração de energia elétrica. I. Título.

CDD 621.3

LUCAS TAVARES DA SILVA

APRENDIZADO DE MÁQUINA APLICADO NA PREVISÃO DA GERAÇÃO DE ENERGIA ELÉTRICA DE UMA USINA SOLAR FOTOVOLTAICA

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia Elétrica do Centro de Tecnologia da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Engenharia Elétrica.

Aprovado em: 10 de Fevereiro de 2022

BANCA EXAMINADORA

Profª. PhD. Ruth Pastôra Saraiva Leão (Orientadora)
Universidade Federal do Ceará (UFC)

Eng. André Wagner de Barros Silva (Coorientador)
Universidade Federal do Ceará (UFC)

Me. Erick Costa Bezerra
Universidade Federal do Ceará (UFC)

À minha mãe, por seu carinho, cuidado e dedicação em todos os momentos. És o meu maior exemplo de fé e serei eternamente grato por ter sempre caminhado ao meu lado, me dando forças, especialmente nos momentos mais difíceis.

AGRADECIMENTOS

A Deus, por ser o meu refúgio e fiel amigo ao longo de toda a minha vida, sobretudo no decurso da graduação.

À minha mãe Elienir, por ser a minha maior referência e estar sempre ao meu lado, sendo a maior investidora em minha educação. À minha avó Edite, pelo carinho e por ser um exemplo de força.

À Universidade Federal do Ceará, em especial aos professores do curso de engenharia elétrica, pela dedicação no oferecimento de uma formação qualificada, pautada nos três pilares universitários. Notadamente, agradeço à minha orientadora profª. PhD. Ruth Pastôra Saraiva Leão pelo apoio, solicitude e todos os ensinamentos.

À banca examinadora, pelas contribuições com este trabalho, bem como disponibilidade na avaliação do mesmo.

Aos colegas do Centro de Empreendedorismo da UFC (CEMP), por serem parte fundamental da melhor experiência universitária que tive, e aos colegas de turma, pela cooperação e aprendizados compartilhados.

"Para mim, a determinação consiste em uma combinação entre a vontade de trabalhar duro, a força emocional, uma grande capacidade de concentração e a recusa em admitir a derrota."

(Sir Alex Ferguson)

RESUMO

Tendo em vista a crescente inserção de geração solar fotovoltaica nas matrizes energéticas brasileira e mundial e considerando que a geração proveniente de fonte solar se caracteriza pela intermitência, surge, então, a necessidade de se ter modelos precisos para previsão da geração de energia solar, de modo a permitir melhor planejamento e operação da planta e do sistema elétrico como um todo. Desta forma, neste trabalho, é proposta a aplicação de Aprendizado de Máquina para a previsão da geração de uma usina solar fotovoltaica, de 160 MW de potência instalada, localizada no estado do Ceará. É considerada a tarefa de prever a geração futura, com granularidade horária, para um intervalo de 365 dias (um ano), a partir do conhecimento (treinamento) da série histórica de geração de energia e dados meteorológicos coletados na usina, também abrangendo o período de um ano. Para tal, foram implementados 13 diferentes modelos de previsão, distintos entre si em metodologia e/ou método de treinamento. Os modelos computacionais implementados no trabalho abrangeram abordagens de reconhecimento de sequências, Redes Neurais Artificiais (RNAs) e *XGBoost*, além de métodos híbridos, originados da combinação de dois métodos distintos. Os modelos implementados tiveram, então, suas performances avaliadas a partir de métricas de desempenho (erro frente à geração real), onde verificou-se qual modelo apresentou melhor previsão, a saber, o algoritmo *XGBoost*. A principal contribuição do trabalho foi a identificação dos modelos com melhor desempenho na tarefa proposta, inclusive quando considerada a indisponibilidade de dados meteorológicos.

Palavras-chave: Aprendizado de Máquina. Energia Solar. Inteligência Artificial. Previsão. Geração de Energia Elétrica.

ABSTRACT

Given the growing integration of photovoltaic (PV) solar power generation in Brazilian and world power grids and considering that solar power generation is intermittent, there is an urge to build prediction models to forecast the solar power output, in order to allow better planning and operation of the power plant and the electrical grid itself. Thus, this paper proposes the application of Machine Learning to forecast the power generation of a 160 MW PV plant located in the state of Ceará. It is proposed the task of forecasting the PV power output at hourly intervals for the next 365 days (one year), given the previous year's time series of PV power outputs and weather data collected from the plant. To this end, 13 distinct prediction models were implemented, being different from each other in either methodology or training method. The models implemented in this paper are based on: sequences recognition, Artificial Neural Networks (ANNs), and XGBoost, as well as there is the use of hybrid models, coupling two distinct methods. Then, the implemented models had their performance evaluated using error-based metrics, which showed that the XGBoost model achieved the most accurate results. The main contribution of this work was the identification of the models with the best performance in the proposed task, including the scenario in which the weather data was unavailable.

Keywords: Machine Learning. Solar Energy. Artificial Intelligence. Forecast. Power generation.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

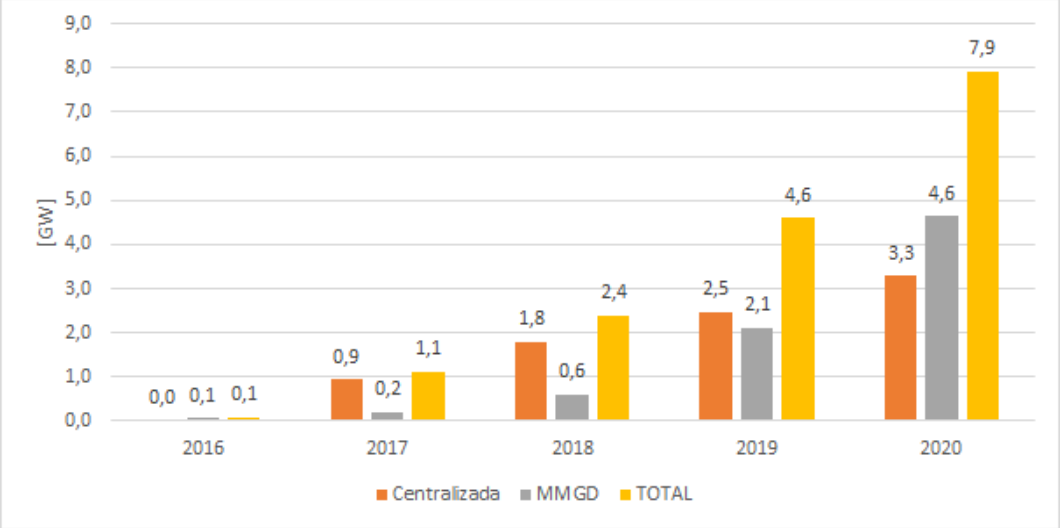
AdaBoost	*Adaptive Boosting*
ARIMA	*Autoregressive Integrated Moving Average* / Média Móvel Integrada Autorregressiva
ARMA	*Autoregressive Moving Average* / Média Móvel Autorregressiva
CV	*Cross Validation* / Validação Cruzada
DL	*Deep Learning* / Aprendizagem Profunda
DT	*Decision Tree* / Árvore de Decisão
GD	Geração Distribuída
i.e.	*id est* / isto é
IA	Inteligência Artificial
IC	Inteligência Computacional
MAE	*Mean Absolute Error* / Erro Absoluto Médio
ML	*Machine Learning* / Aprendizado de Máquina
MLP	*Multilayer Perceptron* / Perceptron Multicamadas
MMGD	Micro e Minigeração Distribuída
PSF	*Pattern Sequence-based Forecast* / Previsão baseada em Sequência de Padrões
PSNN	*Pattern Sequence Neural Network* / Rede Neural em Sequência de Padrões
Q-Q	Gráfico Quantil-Quantil
ReLU	*Rectified Linear Unit* / Unidade Linear Retificada
RF	*Random Forest* / Floresta Aleatória
RMSE	*Root Mean Squared Error* / Raiz do Erro Quadrático Médio
RNA	Rede Neural Artificial
RNC	Rede Neural Convolucional
SBMO	*Root Mean Squared Error* / Sequential Model-Based Optimization
SGD	*Stochastic Gradient Descent* / Gradiente Descendente Estocástico
SVM	*Support Vector Machine* / Máquina de Vetores de Suporte
SVR	*Support Vector Regression* / Regressão por Vetores Suporte
tanh	Tangente Hiperbólica
XGBoost	*Extreme Gradient Boosting*

SUMÁRIO

1 INTRODUÇÃO

A geração de energia solar fotovoltaica tem ganhado cada vez mais inserção nas matrizes energéticas mundial e nacional, com crescimentos da capacidade instalada de aproximadamente 84% e 603% - respectivamente - entre 2017 e 2020, de modo que, ao final de 2020, a capacidade instalada total mundial correspondia a 707,5 GW, com o Brasil tendo capacidade igual a 7,9 GW (EPE, 2021; BP, 2021). A Figura 1 apresenta a evolução da capacidade instalada no Brasil nos últimos anos.

Figura 1 – Evolução da capacidade instalada de geração solar fotovoltaica no Brasil



Fonte: o próprio autor, com dados de: (EPE, 2021).

Levantamentos mais recentes apontam que, em 2022, o país já superou a marca de 13 GW de capacidade instalada de geração solar fotovoltaica (HEIN, 2022). Esse crescimento se deve em grande parte à ampla integração de Micro e Minigeração Distribuída (MMGD) no país, porém, usinas centralizadas de médio e grande porte também apresentam grande potencial de aplicação no Brasil (PEREIRA *et al.*, 2017), sendo que ao final de 2021 o país atingiu a marca histórica de 4 GW de capacidade instalada em usinas solares fotovoltaicas de grande porte (RIBEIRO, 2021). A Figura 2 mostra um exemplo de usina solar fotovoltaica centralizada de grande porte, mais especificamente a usina de São Gonçalo (PI), maior parque solar da América do Sul, com 608 MW de potência operacional. É possível observar os painéis instalados em estruturas montadas no solo, como é usual nas usinas de maior porte.

Figura 2 – Exemplo de usina solar fotovoltaica centralizada de grande porte – Parque Solar São Gonçalo (PI)

Fonte: (ENEL GREEN POWER, 2021).

Esse elevado crescimento da geração solar fotovoltaica se justifica sobretudo em razão desta ser uma fonte de geração limpa, renovável e de fácil aplicação para Geração Distribuída (GD) e geração centralizada, além dos diversos avanços ao longo dos anos nas tecnologias relativas à construção, operação e manutenção das usinas solares, bem como a constante redução de custos associados a esse tipo de geração (NWAIGWE *et al.*, 2019; OLIVER; JACKSON, 2001). No entanto, o fato da geração solar se caracterizar como uma fonte intermitente traz consigo desafios. Uma vez que a geração é dependente de fatores meteorológicos, como irradiância solar, temperatura ambiente, ocorrências de chuvas, vento, poeira, entre outras, tem-se que a produção de energia pode ser altamente variável, de modo que a integração da geração solar na rede elétrica pode ser dificultada. Nesse sentido, surge a necessidade do conhecimento da geração futura de determinada usina (ou conjunto de usinas) de maneira precisa e confiável (RAZA *et al.*, 2016). Uma previsão segura da geração para dado intervalo futuro tende a diminuir incertezas, aumentando a estabilidade e a viabilidade econômica do sistema (DIAGNE *et al.*, 2013).

Diante do exposto, este trabalho propõe-se a implementar e avaliar o desempenho de diversos modelos computacionais para a previsão da geração de uma usina solar fotovoltaica, conforme objetivos mencionados adiante.

1.1 Objetivos

Este trabalho tem como principal objetivo a seleção de um algoritmo computacional para previsão de geração solar fotovoltaica para um horizonte de 24h à frente.

Os objetivos específicos do trabalho são:

– Aplicar e avaliar o desempenho de 13 (treze) métodos distintos, nas categorias de reconhecimento de sequências, aprendizagem supervisionada e híbridos, para a previsão da geração fotovoltaica.

– Avaliar a relevância de dados meteorológicos no desempenho dos métodos de previsão.

1.2 Metodologias de previsão

Os modelos de previsão de geração fotovoltaica são em geral classificados em três categorias: modelos físicos, estatísticos e híbridos. Os métodos físicos usam um modelo de simulação teórica para calcular a potência de saída de um sistema fotovoltaico com base em seus principais parâmetros de projeto. Os modelos estatísticos incluem todos os métodos baseados em dados, cobrindo tanto a modelagem estatística clássica, os modelos analíticos, quanto os modelos no campo da Inteligência Computacional (IC). O método híbrido é uma combinação de dois métodos diferentes, um físico e um estatístico, ou dois ou mais métodos estatísticos (MAYER; GRÓF, 2021). Os métodos estatísticos são mais comumente usados para previsão de geração fotovoltaica (ANTONANZAS *et al.*, 2016). Esses métodos orientados a dados são baseados em conjuntos de dados históricos de irradiância e produção de energia, e eles não exigem qualquer informação sobre os parâmetros de projeto do sistema fotovoltaico. Os modelos de previsão no campo da Inteligência Computacional são inúmeros e incluem as Redes Neurais Artificiais (RNAs), algoritmos de aprendizado profundo, máquinas de aprendizado e algoritmos para reconhecimentos de padrões (AHMED *et al.*, 2020). Em (SOBRI *et al.*, 2018), os autores destacam que, dentre estas categorias citadas, os métodos utilizados fazem uso principalmente de: Média Móvel Autorregressiva (*Autoregressive Moving Average* – ARMA), Média Móvel Integrada Autorregressiva (*Autoregressive Integrated Moving Average* – ARIMA), regressão linear, Regressão por Vetores Suporte (*Support Vector Regression* – SVR), Máquina de Vetores de Suporte (*Support Vector Machine* – SVM), Árvores de Decisão (*Decision Trees* – DT) e RNAs.

Os modelos estatísticos analíticos são aplicações mais consolidadas, tendo em vista já serem amplamente utilizados para previsões nas mais diversas áreas do conhecimento (MAJID; MIR, 2018; MAKRIDAKIS *et al.*, 2018). Dentre estes, os modelos de persistência, que assumem que a geração futura será igual à geração passada, são os mais simples de serem construídos, sendo amplamente utilizados como modelos de referência para *benchmark* (CHU *et al.*, 2021). Dada esta característica de simplicidade, a adoção de modelos que não o de persistência só é justificável se houver ganho de precisão. Nesse sentido, as abordagens de Inteligência Computacional têm ganhado notoriedade nos últimos anos, exatamente pelo potencial de produzirem previsões mais precisas e de maneira rápida (KALOGIROU; SENCAN, 2010).

Recentemente, em (WANG *et al.*, 2017) foi proposta a utilização de modelos de Previsão baseada em Sequência de Padrões (*Pattern Sequence-based Forecast* – PSF) para a previsão de geração de usinas solares, enquanto que Lin *et al.* (2019) introduziram uma abordagem híbrida pela utilização de PSF combinada a RNAs. Em ambos os casos, os resultados foram promissores e instigaram maior investigação da aplicação destes métodos no tema proposto.

Além disso, há trabalhos recentes que visaram outras abordagens de Aprendizagem Profunda (*Deep Learning* – DL) além das redes neurais, sobretudo árvores de decisões e *Extreme Gradient Boosting* (*XGBoost*), para previsões de geração eólica (MACHADO *et al.*, 2021) e de irradiância solar (KUMARI; TOSHNIWAL, 2021; HEINEN, 2018; KAMAROUTHU, 2020). Nesse sentido, este trabalho avalia treze diferentes modelos de previsão, que serão melhor detalhados nos capítulos seguintes.

1.3 Estrutura do trabalho

Este trabalho é constituído de seis capítulos. O capítulo 1 apresenta uma visão do crescimento da geração solar fotovoltaica no Brasil e os objetivos do trabalho. O capítulo 2 detalha as características da base de dados utilizada ao longo do trabalho, bem como discorre sobre os tratamentos iniciais aplicados a esta base (pré-processamento dos dados), necessários para continuidade (e reprodutibilidade) do trabalho. O capítulo 3 apresenta os diferentes algoritmos computacionais adotados para desenvolvimento no trabalho, detalhando suas respectivas metodologias e estruturas. O capítulo 4 detalha a implementação computacional dos algoritmos de previsão, principais objetos de estudo do trabalho. O capítulo 5 apresenta e discute os resultados obtidos, avaliando qual algoritmo obteve melhor desempenho. Finalmente, o capítulo 6 aborda as conclusões obtidas e as proposições de trabalhos futuros.

2 ESTRUTURA E PRÉ-PROCESSAMENTO DOS DADOS

Tendo em vista os objetivos citados na seção 1.1, este trabalho fez uso de dados históricos (séries temporais) de geração de energia elétrica e grandezas meteorológicas para aplicação nos algoritmos de previsão. Os dados foram coletados em uma usina solar fotovoltaica, de 160 MW de capacidade instalada, localizada na região leste do estado do Ceará. A seção a seguir detalha como estes dados foram estruturados para utilização no trabalho. Na seção seguinte é abordado o pré-processamento dos dados, que consistiu nos testes de normalidade, detecção de dados anômalos (*outliers*), normalização dos dados e clusterização das bases.

2.1 Estrutura dos dados

Os dados utilizados neste trabalho contemplaram o período de dois anos, entre 30 de junho de 2019 e 29 de junho de 2021 (731 dias), e consistiram em: (1) série temporal de geração de energia da usina, com granularidade horária e (2) série temporal de dados meteorológicos, coletados na usina. A escolha dos atributos (*features*), bem como a própria estruturação dos dados em si, foi motivada visando maior conformidade com os algoritmos PSF implementados, conforme (LIN *et al.*, 2019).

A Tabela 1 apresenta a estrutura, com especificação das *features* utilizadas, bem como nomenclaturas adotadas para os conjuntos criados.

Tabela 1 – Estrutura dos dados para os algoritmos PSF

Origem dos dados	Conjunto de *features*	Informações dos dados
Dados de geração (*PV data*)	$PV \in \mathbb{R}^{731\times12}$	Dados de geração entre 5:00 e 17:00, com granularidade horária (12 dados por dia).
Dados meteorológicos completos	$W1 \in \mathbb{R}^{731\times10}$	Dados diários: temperaturas ambiente mínima e máxima, precipitação e irradiância média; às 9:00 e 15:00: temperatura ambiente, umidade relativa do ar e velocidade do vento.
Dados meteorológicos reduzidos	$W2 \in \mathbb{R}^{731\times4}$	Dados diários: temperaturas ambiente mínima e máxima, precipitação e irradiância média; *W2* é um subconjunto de *W1*.

Fonte: o próprio autor.

Para a implementação dos algoritmos de aprendizagem supervisionada (seções 4.2 e 4.3), que necessitam de entradas (X) associadas a saídas (y), os dados da Tabela 1 foram adaptados da seguinte forma: para a aprendizagem com utilização apenas dos dados de geração

de energia, fez-se a associação de cada dia (X) para seu respectivo dia seguinte (y), em um deslocamento (*shift*) unitário (um dia), conforme em (BROWNLEE, 2016). Em decorrência dessa adaptação, o número total de dias disponíveis foi reduzido em um, restando um conjunto com 730 dias. Para a aprendizagem com utilização de todos os dados (geração de energia e meteorológicos), a adaptação consiste apenas em tornar a base meteorológica (*W1*) na entrada (X) e os dados de geração na respectiva saída (y); esta associação não ocasiona perda de dados. A Tabela 2 resume a estrutura mencionada.

Tabela 2 – Estrutura dos dados para aprendizagem supervisionada

Origem dos dados	Conjunto de *features*	Informações dos dados
Dados de geração (*PV data*)	$X \vert y \in \mathbb{R}^{730\times12}$	X: dados de geração entre 5:00 e 17:00, com granularidade horária (12 dados por dia); y: dados de geração dos respectivos dias seguintes.
Dados de geração + dados meteorológicos completos	$X \in \mathbb{R}^{731\times10}$ $\mathrm{y} \in \mathbb{R}^{731\times12}$	X: dados meteorológicos completos; y: dados de geração entre 5:00 e 17:00, com granularidade horária (12 dados por dia).

Fonte: o próprio autor.

2.2 Pré-processamento dos dados

A etapa de pré-processamento dos dados consistiu nos tratamentos aplicados no conjunto de dados de modo a prepará-los para utilização nos algoritmos de previsão (capítulo 4). Todas essas etapas antecederam a implementação dos algoritmos de previsão e são de extrema importância para o correto funcionamento dos mesmos. As seções a seguir detalham os testes de normalidade – que verificaram se os dados utilizados no trabalho apresentam distribuição normal –, a detecção de dados anômalos (*outliers*), a normalização dos dados e as clusterizações realizadas nos conjuntos de dados.

2.2.1 Testes de normalidade

Os testes de normalidade se fizeram necessários em razão de que muitos modelos estatístico-matemáticos assumem que os dados apresentam distribuição Gaussiana (distribuição normal), de modo que, em caso de utilização destes modelos com dados que não atendem a essa premissa, os resultados podem apresentar incoerências (BROWNLEE, 2019). Nesse sentido, verificou-se se os dados utilizados no trabalho apresentam distribuição normal para que os modelos utilizados fossem coerentes com o conjunto de dados.

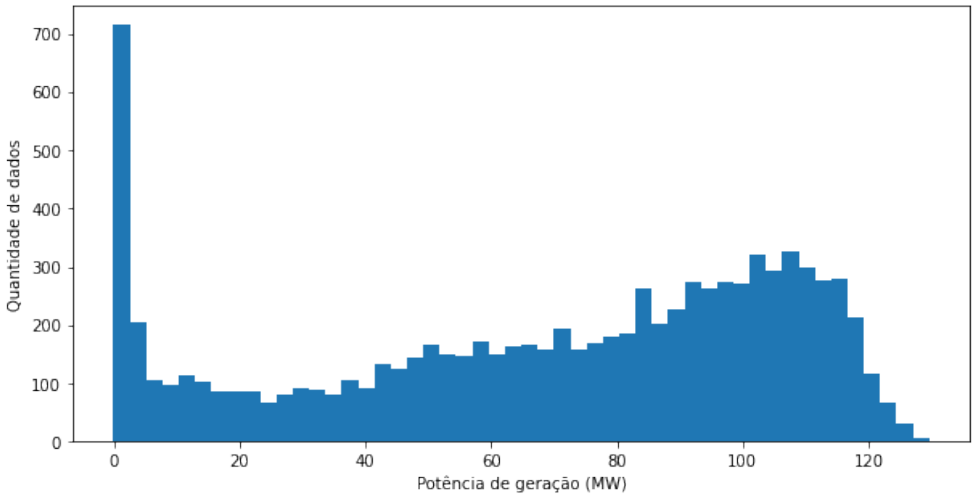
Os principais testes de normalidade presentes na literatura são o histograma, Gráfico Quantil-Quantil (Q-Q), Teste de Shapiro-Wilk e Teste D'Agostino-Pearson (DAS; IMON, 2016; GHASEMI; ZAHEDIASL, 2012). Assim, foram executados cada um dos testes para os dados de geração de energia, onde verificou-se que todos apresentaram resultados compatíveis. Para os dados meteorológicos, em razão da quantidade de atributos (dez), optou-se apenas pela aplicação do Teste D'Agostino-Pearson, já que os métodos visuais seriam de verificação mais exaustiva.

2.2.1.1 Testes de normalidade dos dados de geração de energia

Primeiramente, foi feita a verificação do histograma dos dados de geração de energia. Um histograma consiste numa representação visual de dados quantitativos e permite visualizar se os mesmos possuem similaridade com a curva Normal (SIQUEIRA, 2021).

A Figura 3 apresenta o histograma dos dados de geração utilizados no trabalho. É fácil notar que a representação não apresenta similaridade com a curva Normal.

Figura 3 – Histograma dos dados de geração de energia da usina

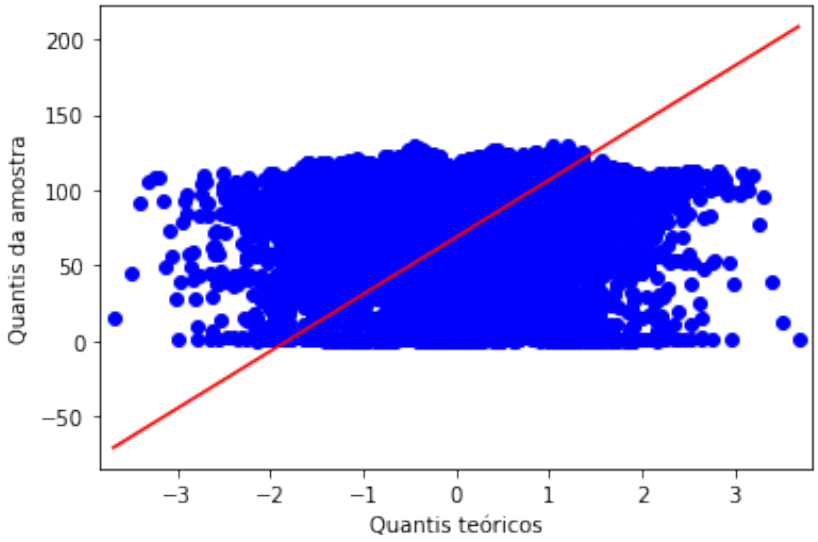


Fonte: o próprio autor.

O Gráfico Quantil-Quantil Q-Q é um gráfico de probabilidades que permite comparar duas distribuições de probabilidade; assim, trançando a distribuição de probabilidade do conjunto de dados em estudo, juntamente com a da distribuição normal, pode-se verificar se estes apresentam o comportamento linear verificado na distribuição normal; caso não apresentem, estes não têm característica normal (DAS; IMON, 2016).

A Figura 4 apresenta o Gráfico Quantil-Quantil dos dados de geração utilizados no trabalho. Verificou-se que os dados estão deveras distante de um comportamento linear, de modo que não são característicos de distribuição normal, em conformidade com o teste anterior.

Figura 4 – Gráfico Quantil-Quantil dos dados de geração de energia da usina



Fonte: o próprio autor.

O Teste de Shapiro-Wilk foi proposto em 1965 e avalia a hipótese nula de que a amostra em questão (conjunto de dados submetidos ao teste) tem distribuição normal; o teste retorna um valor de estatística W associada a uma significância (valor-p) que, caso seja superior a 0,05 indica que a amostra tem comportamento normal (SHAPIRO; WILK, 1965).

Para o conjunto de dados de geração de energia da usina, o valor de W retornado pelo teste foi de 0,919 com valor-p 0,0001, indicando que os dados não apresentam distribuição normal, em conformidade com os testes anteriores.

Finalmente, também foi executado o Teste D'Agostino-Pearson, proposto em 1970, que também avalia a hipótese nula de que a amostra tem distribuição normal; o teste retorna um valor de estatística D associada a uma significância (valor-p) que, caso seja superior a 0,05 indica que a amostra tem comportamento normal. Sua utilização neste trabalho justifica-se por este apresentar melhor desempenho para grandes amostras (YAP; SIM, 2011), o que é o caso deste trabalho.

Para o conjunto de dados de geração de energia da usina, o valor de D retornado pelo teste foi de 2668,78 com valor-p 0,0001, indicando que os dados não apresentam distribuição normal, em conformidade com os testes anteriores.

2.2.1.2 Testes de normalidade dos dados meteorológicos

Tendo em vista as razões supracitadas, foi executado apenas o Teste D'Agostino-Pearson para verificação da normalidade dos dados meteorológicos. A Tabela 3 apresenta os valores de D e seus respectivos valores-p obtidos, para cada uma das grandezas. Haja vista todos os valores-p abaixo de 0,05, nenhuma das séries apresenta comportamento normal.

Tabela 3 – Resultados do teste de normalidade dos dados meteorológicos

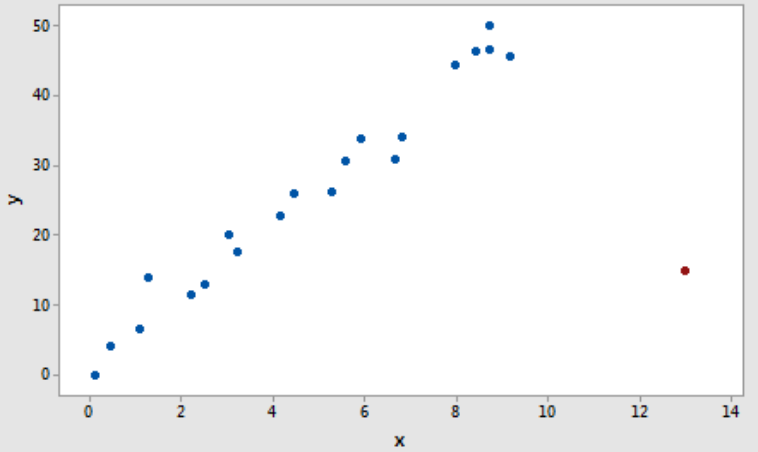
Grandeza	D	Valor-p
Temperatura ambiente mínima	365,563	0,001
Temperatura ambiente máxima	31,412	0,001
Precipitação	822,695	0,001
Irradiância média	144,981	0,001
Temperatura ambiente - 9:00	11,883	0,001
Temperatura ambiente - 15:00	154,092	0,001
Umidade relativa do ar- 09:00	27,681	0,001
Umidade relativa do ar - 15:00	52,855	0,001
Velocidade do vento - 09:00	7,811	0,001
Velocidade do vento - 15:00	277,367	0,001

Fonte: o próprio autor.

2.2.2 Detecção de outliers

Em continuidade aos tratamentos à base de dados, foi feita a detecção de *outliers*, ou dados anômalos, que são aqueles que fogem do comportamento geral dos dados, levando a crer que são dados inconsistentes (HAWKINS, 1980). A Figura 5 apresenta um exemplo de *outlier* (em vermelho) em um conjunto de dados (em azul).

Figura 5 – Exemplo de *outlier*



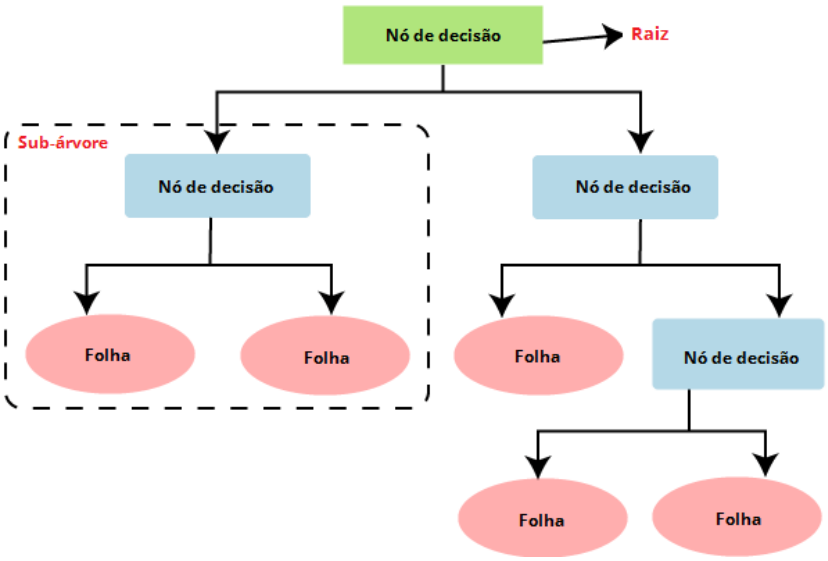
Fonte: adaptado de: (EBERLY COLLEGE OF SCIENCE, 2021?).

A identificação e o tratamento de *outliers* é deveras importante no contexto da preparação dos dados, tendo em vista que estes, se presentes nos dados de treinamento dos modelos, podem introduzir viés indesejado nos algoritmos computacionais, distorcendo os resultados finais. Para que os resultados sejam consistentes, a base de dados também deve ser e *outliers* devem estar ausentes (SMITI, 2020).

A detecção de *outliers* empregada neste trabalho foi feita utilizando o algoritmo de floresta de isolamento. Visando o entendimento adequado do conceito do mesmo, primeiro discute-se, brevemente, os conceitos de árvores de decisão e florestas aleatórias.

As árvores de decisão (DT) são modelos não paramétricos constituídos de grafos acíclicos nos quais cada nó é um nó de decisão (*decision node*), com dois ou mais nós sucessores, ou uma folha (*leaf node*), este último também denominado nó-terminal. Em termos gerais, as árvores de decisão são funções de decisão que dividem os dados de acordo com suas características (atributos) e com determinados parâmetros (regras), de modo que um problema complexo seja divido em subproblemas mais simples e, recursivamente, a mesma estratégia seja aplicada aos subproblemas até a solução do problema como um todo (GAMA *et al.*, 2004). A Figura 6 mostra uma representação simplificada de uma árvore de decisão, com quatro nós de decisão e quatro folhas (nós-terminais).

Figura 6 – Representação de uma árvore de decisão



Fonte: traduzido de: (ARAIN, 2021?).

O conceito de Florestas Aleatórias (*Random Forests* – RF) é facilmente entendido a partir do conhecimento das árvores de decisão, uma vez que uma floresta aleatória é constituída exatamente de diversas árvores de decisão (daí o nome *floresta*). Cada árvore é treinada com uma

amostra dos dados e procede com suas decisões, conforme supracitado, gerando seus respectivos resultados; a predição final da floresta é, então, determinada a partir dos resultados das árvores, seja por uma média ou voto ponderado (MOISEN, 2008). Em razão disto, as florestas aleatórias apresentam maior precisão do que árvores aleatórias, preservando a robustez a *outliers* e ruídos (HASTIE *et al.*, 2009).

Finalmente, o modelo de floresta de isolamento consiste em dividir determinado conjunto de dados em duas partes (realizando uma cisão) e seguir recursivamente dividindo o conjunto, utilizando árvores de decisão, até que cada ponto do conjunto esteja isolado (seja um nó-terminal). Após isso, são determinados os tamanhos dos caminhos (profundidade) desde cada ponto até a raiz, ou nó inicial (LESOUPLE *et al.*, 2021). Na detecção de anomalias (*outliers*), os pontos anômalos serão aqueles com os menores caminhos (menor profundidade), significando que estarão demasiadamente isolados, visto que *outliers* tendem a se isolar após poucas divisões, enquanto que os pontos não anômalos terão caminhos maiores, característicos de pontos em regiões de maior densidade (MENSI; BICEGO, 2021).

A utilização de floresta de isolamento para detecção de *outliers* neste trabalho se deu em razão deste método apresentar alta acurácia, especialmente em grandes bases de dados, aliada a elevada eficiência computacional (LIU *et al.*, 2008). Além disso, trata-se de um método não paramétrico, o que se mostrou necessário neste trabalho, tendo em vista a não normalidade dos dados utilizados (seção 2.2.1).

Assim, no âmbito da execução da detecção de *outliers*, utilizou-se o algoritmo presente na biblioteca Python *Scikit-learn*. O algoritmo foi primeiramente aplicado aos dados meteorológicos, configurado com contaminação 5% e 5000 estimadores, valores usuais na literatura (LIU *et al.*, 2008) e retornou a identificação de 37 dias anômalos, que foram tratados pelo método do vizinho mais próximo.

Em seguida, foi criado um novo modelo de Floresta de Isolamento, configurado com contaminação 1% (haja vista base de dados menos diversificada, pois só contém dados de potência/energia) e 5000 estimadores; o algoritmo, então, identificou 8 dias anômalos, que também foram tratados pelo método do vizinho mais próximo, identificando o dia mais similar a partir da base de dados meteorológicos e fazendo a correspondência de dados de geração.

2.2.3 Normalização dos dados

A normalização dos dados consiste em fazer com que todos os atributos (*features*) do conjunto de dados possuam a mesma escala, de modo que os modelos computacionais não recebam dados já observando "pesos" ou preferência a determinados atributos (GÉRON, 2019). Por exemplo, na base de dados meteorológicos há tanto atributos de temperatura ambiente, que em geral estão entre 20 ºC e 40 ºC, como de velocidade do vento, em geral não superiores a 10 m/s. Dessa forma, em caso de não normalização dos dados, os algoritmos poderiam entender os atributos de temperatura ambiente como mais proeminentes, pois são numericamente maiores, o que não necessariamente é verdade.

Assim, implementou-se o escalonamento mínimo-máximo para normalização dos dados. Este método consiste basicamente em redimensionar os dados para que fiquem no intervalo [0,1]. Isso é feito subtraindo o valor mínimo e dividindo pelo máximo menos o mínimo; dessa forma, o valor mínimo do conjunto de dados é redimensionado para 0 e o valor máximo, para 1, com todos os demais dados dentro desse intervalo.

A implementação computacional do escalonamento mínimo-máximo fez uso da classe *MinMaxScaler*, que integra a biblioteca de código aberto *Scikit-learn*, em linguagem de programação Python, amplamente utilizada neste trabalho.

Havia uma pequena quantidade de valores faltantes (1,19%) nos dados meteorológicos, que foram tratados por uma abordagem de vizinho mais próximo, conforme (WANG *et al.*, 2017), em que se um dia d, com vetor de dados meteorológicos W^d, possui dados faltantes, localiza-se seu vizinho mais próximo (dia s, com vetor W^s) por meio da distância Euclidiana e os valores disponíveis em W^d; os dados faltantes em W^d são, então, substituídos pelos respectivos valores em W^s. Estes tratamentos foram feitos após a detecção de *outliers* e normalização dos dados (seções 2.2.2 e 2.2.3).

2.2.4 Clusterização dos dados

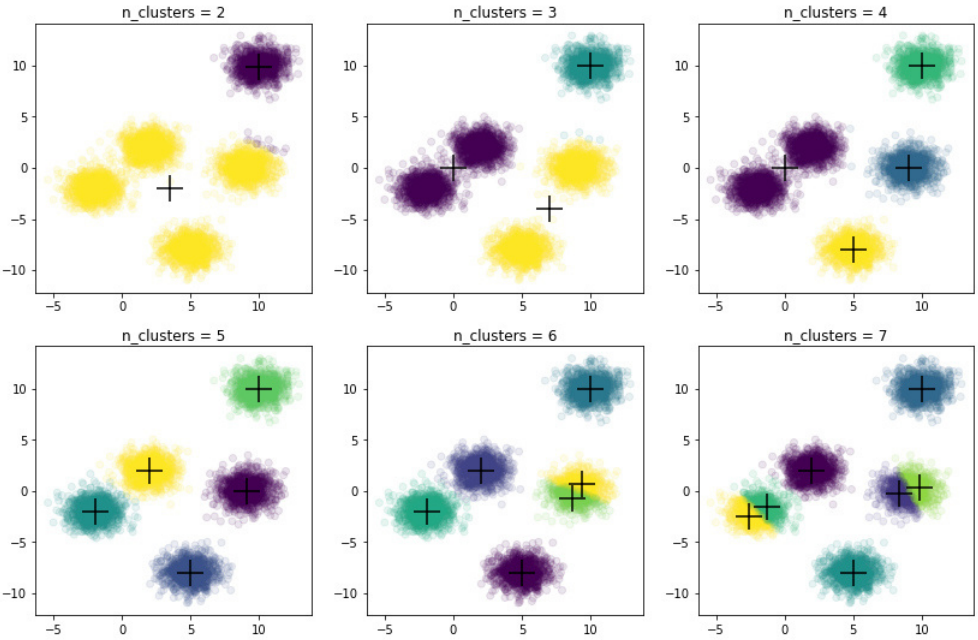
A clusterização consiste no agrupamento de um conjunto de objetos em uma determinada quantidade de *clusters* (grupos), com base nas características que estes objetos possuem (GRUS, 2016). O objetivo é reunir os objetos de tal forma que sejam altamente similares dentro de um mesmo *cluster* e que diferentes *clusters* tenham baixa similaridade entre si (XU; WUNSCH, 2005). Como cada *cluster* terá seu respectivo centro (centroide), os estudos de-

senvolvidos no conjunto de objetos podem, por exemplo, levar em consideração apenas os centroides, diminuindo significativamente a quantidade de objetos analisados, porém preservando os resultados gerais, já que o centroide se caracteriza como uma espécie de representante do grupo.

Outra abordagem possível, e amplamente utilizada nesse trabalho, é uma vez determinados os *clusters* com base num conjunto inicial de objetos, inferir determinado grupo para um dado novo objeto, identificando qual o *cluster* mais similar a este. Tendo em vista que o trabalho se propõe a prever a geração para dias futuros com base em dados históricos, alguns dos algoritmos implementados fazem uso dessa abordagem de clusterização, como será detalhado no capítulo 4.

A Figura 7 apresenta um exemplo de seis diferentes clusterizações para um mesmo conjunto de dados, com o número de *clusters* variando entre 2 e 7; os *clusters* são diferenciados pelas colorações, enquanto que as marcações em cruz indicam os centroides de cada *cluster*. Por uma análise visual, é possível verificar que o a divisão em 5 *clusters* apresentou a clusterização mais adequada, com os subconjuntos completamente separados entre si.

Figura 7 – Exemplo de diferentes clusterizações de um mesmo conjunto de dados, a partir da variação do número de *clusters*



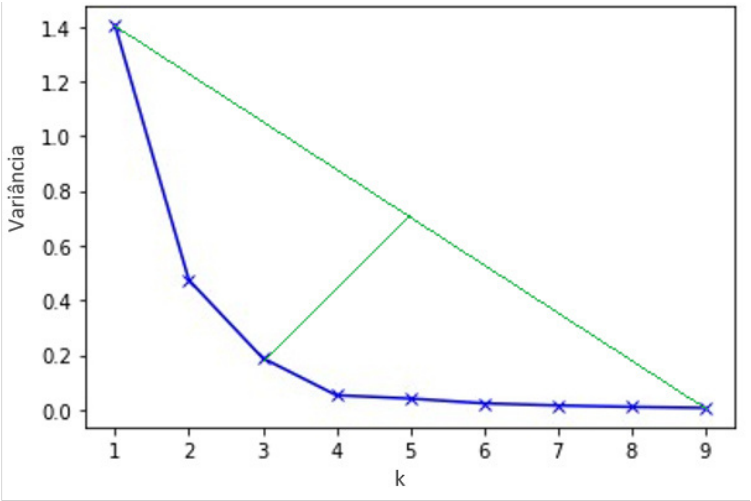
Fonte: o próprio autor.

Para implementação da clusterização, foi adotada a técnica K-*means*, ou K-médias, em razão de sua ampla utilização na literatura, conforme (XU; WUNSCH, 2005; AGGARWAL; REDDY, 2014). O algoritmo K-*means* divide um conjunto de *n* objetos em um número pré-

-determinado de *K clusters* de modo que a distância *intra-cluster* (interna aos *clusters*) seja mínima e *inter-cluster* (entre *clusters*) seja máxima (HONDA, 2017). A implementação mais comum do K-*means* posiciona *K* centroides iniciais no conjunto de objetos, atribui objetos para o centroide mais próximo, recalcula o valor do centroide (média dos objetos atribuídos) e repete esse procedimento até convergir (VANDERPLAS, 2017).

Para determinação do parâmetro *K* do algoritmo K-*means*, fez-se uso do Método *Elbow*, ou Método do Cotovelo, que consiste em clusterizar o conjunto de objetos para diferentes valores de *K*, calcular a variância dos dados em relação ao respectivo *K* e observar em gráfico o ponto a partir do qual o aumento do número de *clusters* (*K*) não corresponde a uma redução significativa da variância; esse ponto é identificado por apresentar maior distância em relação à reta entre os centroides extremos e assemelha-se a um cotovelo, daí a nomenclatura do método (BENZAKE, 2018). A figura 8 apresenta um exemplo de determinação do valor de *K* dentro do intervalo [1,9]; pela curva, o valor ideal para *K* é 3.

Figura 8 – Exemplo de determinação do número ótimo de *clusters* por meio do Método *Elbow*



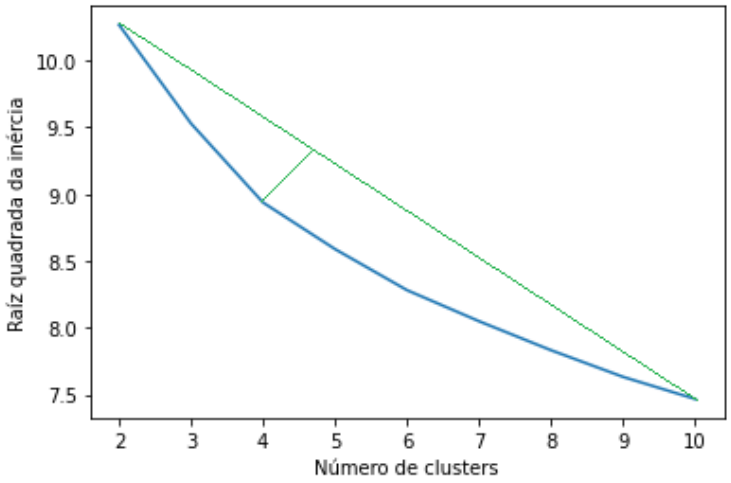
Fonte: adaptado de: (BENZAKE, 2018).

Adicionalmente, como forma de validação do valor de *K* obtido pelo Método *Elbow*, aplicou-se o cálculo da Silhueta (no inglês, *Silhouette*), conforme (MARTÍNEZ-ÁLVAREZ *et al.*, 2011), avaliando se a Silhueta, que estará dentro do intervalo [-1, +1], de fato apresenta bom valor de conformidade (valor positivo, sobretudo próximo a +1) para o *K* selecionado.

Tendo em vista os algoritmos PSF, PSF1, PSF2, bem como algoritmos híbridos, apresentados no capítulo 3, fez-se necessária a clusterização dos conjuntos de dados, de modo que aplicou-se o K-*means* em cada um dos casos.

Aplicando o Método *Elbow* para seleção do parâmetro K da clusterização dos dados de geração (base *PV data*), obteve-se o valor ótimo igual a 4, conforme Figura 9. A silhueta associada ao K igual a 4 foi obtida como igual a 0,24, que, sendo positiva, indicou conformidade adequada do valor de K.

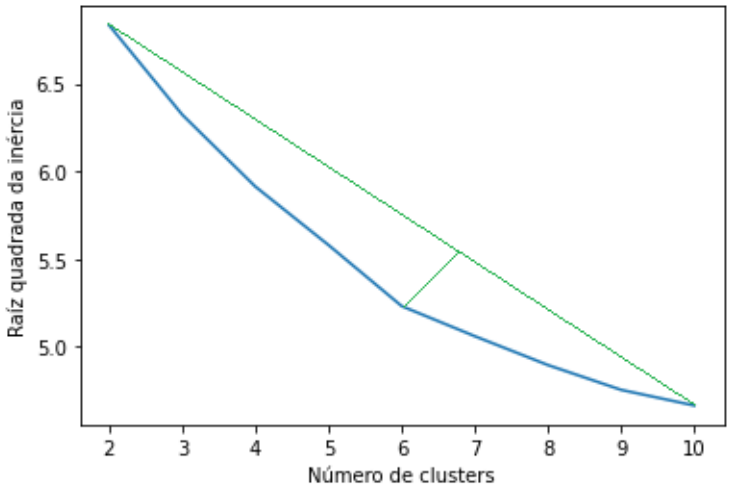
Figura 9 – Determinação do *K*, por meio do Método *Elbow*, para a clusterização dos dados de geração (base *PV data*)



Fonte: o próprio autor.

Analogamente, aplicando o Método do Cotovelo para seleção do parâmetro K da clusterização dos dados meteorológicos completos (base *W1*), obteve-se o valor ótimo igual a 6, conforme Figura 10. A silhueta associada ao K igual a 6 foi obtida como igual a 0,24, que, sendo positiva – de modo similar aos caso anterior –, indicou conformidade adequada do valor de K.

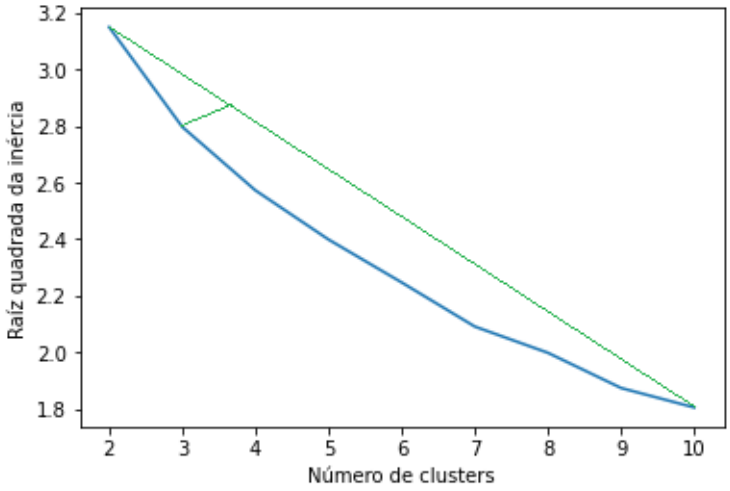
Figura 10 – Determinação do *K*, por meio do Método *Elbow*, para a clusterização dos dados meteorológicos completos (base *W*1)



Fonte: o próprio autor.

Finalmente, aplicando o Método *Elbow* para seleção do parâmetro K da clusterização dos dados meteorológicos reduzidos (base *W2*), obteve-se o valor ótimo igual a 3, conforme Figura 11. A silhueta associada ao K igual a 3 foi obtida como igual a 0,31, que, sendo positiva – de modo similar aos casos anteriores –, indicou conformidade adequada do valor de K.

Figura 11 – Determinação do *K*, por meio do Método *Elbow*, para a clusterização dos dados meteorológicos reduzidos (base *W2*)



Fonte: o próprio autor.

Com os parâmetros K devidamente definidos, as bases de dados *PV data*, *W1* e *W2* foram clusterizadas por meio do método K-*means*, presente na biblioteca *Scikit-learn*. A partir dessas clusterizações, foram construídas séries temporais auxiliares, consistindo nas séries de rótulos de cada dia, i.e., para cada clusterização, uma série de identificadores do *cluster* respectivo a cada dia. Tendo em vista a quantidade de atributos – 12 valores de potência na base *PV data*, 10 grandezas meteorológicas na base *W1* e 4 grandezas meteorológicas na base *W2* – não foi possível traçar uma representação gráfica das clusterizações realizadas, haja vista a incapacidade de representação fiel de um número de dimensões superior a 3.

3 ALGORITMOS COMPUTACIONAIS IMPLEMENTADOS

Este capítulo detalha os diferentes algoritmos computacionais adotados para a previsão da geração fotovoltaica, bem como os algoritmos utilizados na etapa auxiliar de otimização de hiperparâmetros dos modelos. Aqui, deseja-se detalhar os objetivos e conceitos técnicos de cada algoritmo, de modo que suas utilizações sejam devidamente justificadas.

3.1 Reconhecimento de sequências

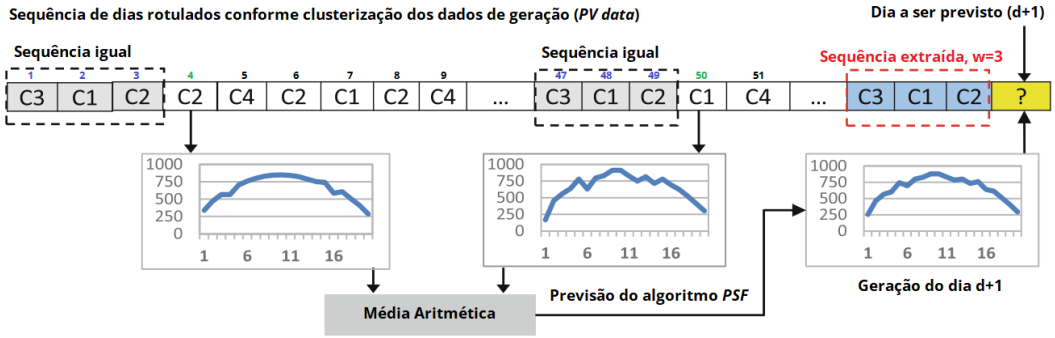
Os primeiros algoritmos para a previsão da geração fotovoltaica (potência de saída da usina) foram baseados no problema de reconhecimento de padrões, que consiste em classificar um conjunto de objetos em determinado número de classes (THEODORIDIS; KOUTROUMBAS, 2003). A partir desta classificação, é possível localizar similaridades e determinar eventuais correlações entre os padrões. As seções 3.1.1 a 3.1.3 detalham o funcionamento dos algoritmos de PSF que foram implementados neste trabalho.

3.1.1 Pattern Sequence-based Forecasting (PSF)

O algoritmo *Pattern Sequence-based Forecasting* (PSF) foi definido em (MARTÍNEZ-ÁLVAREZ *et al.*, 2011) e consiste em, basicamente, três etapas. Inicialmente, os dados históricos (série temporal) de geração de energia (*PV data*) são clusterizados e, para cada dia, é associado um rótulo (*label*) respectivo ao identificador do *cluster* ao qual aquele dia está associado. Após isso, uma janela de tamanho w, imediatamente anterior ao dia para o qual deseja-se realizar a previsão, é extraída da série temporal de rótulos; então, é feita uma busca ao longo da série temporal de rótulos por outras ocorrências da sequência (padrão) da janela extraída. Finalmente, são identificados os dias imediatamente posteriores às demais sequências localizadas e, com os dados de geração correspondentes (base de dados de potência), é calculada uma média entre os dias e esse valor é atribuído como a previsão de geração (potência de saída da usina) para o dia a ser previsto.

A Figura 12 apresenta um esquemático do método PSF. No exemplo, os dados foram clusterizados em 4 *clusters*, de *labels* *C1*, *C2*, *C3* e *C4*; a janela tem tamanho w igual a 3 e, com isso, contém o padrão (*C3*, *C1*, *C2*). A busca na série temporal de rótulos localizou a sequência nos dias 1 a 3 e 47 a 49. Assim, o dia a ser previsto ($d+1$) teve sua geração estimada pela média da geração dos dias 4 e 50, conforme representado na curva em azul.

Figura 12 – Exemplo de aplicação do algoritmo *Pattern Sequence-based Forecasting* (PSF)



Fonte: adaptado de: (WANG *et al.*, 2017).

Com essa construção, o algoritmo contempla previsões para quaisquer horizontes de tempo; para tal, os valores previstos para o primeiro dia futuro ($d+1$) são inseridos na base de dados (série temporal) e a previsão para o segundo dia futuro ($d+2$) segue o mesmo procedimento descrito na Figura 12. O *loop* pode ser tão extenso quanto necessário, abrangendo o número de dias desejado.

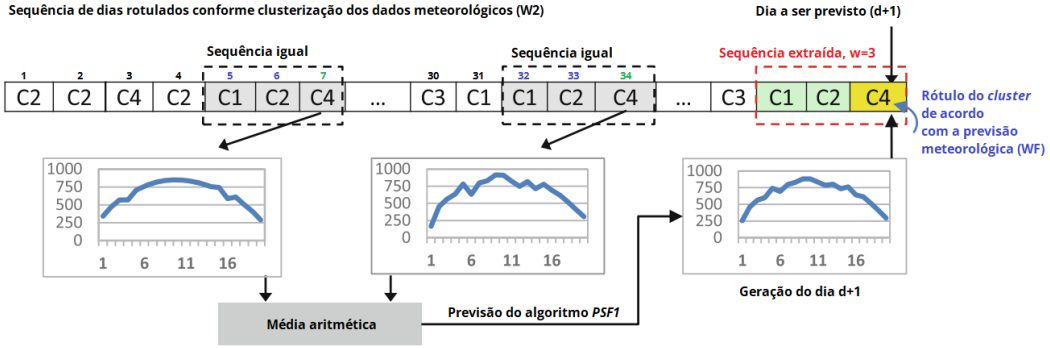
O tamanho ótimo w da janela depende de cada caso (de cada base de dados), porém pode ser ajustado (otimizado) de maneira automática para cada implementação. Neste trabalho, utilizou-se Validação Cruzada (*Cross Validation* – CV), conforme descrita na seção 3.4.4, para otimização do tamanho da janela (w).

3.1.2 Pattern Sequence-based Forecasting 1 (PSF1)

O algoritmo *Pattern Sequence-based Forecasting* 1 (PSF1) é uma extensão do PSF e foi definido em (WANG *et al.*, 2017). Ele se diferencia do PSF na etapa de clusterização, que é feita com a série temporal de dados meteorológicos reduzida (W2). Os dias são clusterizados em k_2 *clusters*, sendo associado a cada dia um rótulo identificador do *cluster* ao qual o dia pertence. Com base na previsão meteorológica (WF) para o dia a ser previsto ($d+1$) é também associado um rótulo para o dia $d+1$, de acordo com os *clusters* definidos inicialmente. Em seguida, uma janela de tamanho w, imediatamente anterior ao dia para o qual deseja-se realizar a previsão (incluindo este), é extraída da série temporal de rótulos (de *clusters*) dos dias. Então, de modo similar ao PSF, é feita uma busca ao longo da série de rótulos por outras ocorrências da sequência (padrão) da janela extraída. Finalmente, são identificados os últimos dias pertencentes a cada janela localizada e, observando os dados de potência correspondentes (base de dados de potência), é feita a previsão da geração (potência de saída da usina) pela média destes dias.

A Figura 13 apresenta um esquemático do método PSF1. No exemplo, os dados (base W2) foram clusterizados em 4 *clusters*, de *labels C1*, *C2*, *C3* e *C4*; a janela tem tamanho w igual a 3 e, com isso, contém o padrão (*C1*, *C2*, *C4*). A busca na série temporal localizou a sequência nos dias 5 a 7 e 32 a 34. Assim, a partir da respectiva base de dados de potência (*PV data*), o dia a ser previsto ($d+1$) teve sua geração estimada pela média da geração dos dias 7 e 34, conforme representado na curva em azul.

Figura 13 – Exemplo de aplicação do algoritmo *Pattern Sequence-based Forecasting* 1 (PSF1)



Fonte: adaptado de: (WANG *et al.*, 2017).

Analogamente ao método PSF, o algoritmo contempla previsões para quaisquer horizontes de tempo, a partir da inclusão da previsão para o dia $d+1$ na base original e seguindo em *loop* com as demais previsões dia-a-dia, abrangendo o número de dias desejado.

O tamanho ótimo w da janela, como no algoritmo anterior, depende de cada implementação. Neste trabalho, a otimização do w foi novamente feita por meio de CV, conforme descrita na seção 3.4.4.

3.1.3 Pattern Sequence-based Forecasting 2 (PSF2)

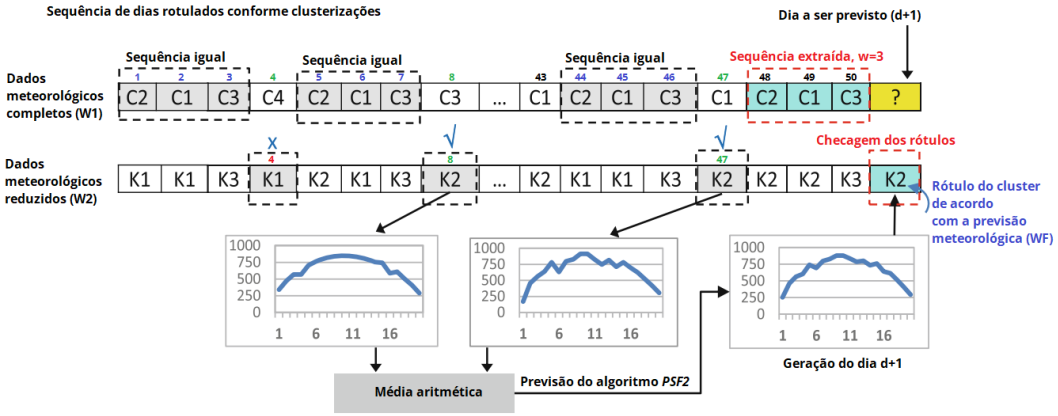
O algoritmo *Pattern Sequence-based Forecasting* 2 (PSF2) é uma segunda extensão do PSF e também foi definido em (WANG *et al.*, 2017). Ele se diferencia dos métodos PSF e PSF1 no que são feitos dois estágios de clusterização. Primeiramente, os dados de treinamento são clusterizados pela série temporal de dados meteorológicos completa (W1) em k_1 *clusters*, sendo a cada dia associado um rótulo relativo ao seu respectivo *cluster*. Em seguida, uma janela de tamanho w, imediatamente anterior ao dia para o qual deseja-se realizar a previsão, é extraída da série temporal de rótulos; então, é feita uma busca ao longo desta mesma série por outras ocorrências da sequência (padrão) da janela extraída. Os dias imediatamente posteriores às

sequências localizadas são considerados “pré-selecionados” para a previsão.

Nesse momento, é observada a segunda clusterização, feita agora a partir da série temporal de dados meteorológicos reduzida (W2). Com os dias clusterizados em k_2 *clusters*, tem-se também uma série temporal de rótulos para essa segunda clusterização. Então, com base na previsão meteorológica (WF) para o dia a ser previsto ($d+1$), é associada uma *label* para o dia $d+1$, de acordo com os *clusters* definidos na segunda clusterização. Finalmente, é verificado quais dias “pré-selecionados” pertencem ao mesmo *cluster* (2ª clusterização) do dia a ser previsto, e a previsão da potência de saída da usina é realizada pela média dos dados de geração destes dias em que há correspondência, descartando eventuais dias “pré-selecionados” cujo *cluster* (2ª clusterização) seja diferente.

A Figura 14 apresenta um esquemático do método PSF2. No exemplo, os dados foram clusterizados, conforme base de dados W1, em 4 *clusters*, de *labels C1*, *C2*, *C3* e *C4*; a janela tem tamanho *w* igual a 3 e, com isso, contém o padrão (*C2*, *C1*, *C3*). A busca na série temporal de rótulos (1ª clusterização) localizou a sequência nos dias 1 a 3, 5 a 7 e 44 a 46, de modo que os dias “pré-selecionados” foram 4, 8 e 47. Conforme a 2ª clusterização – pela base de dados W2 –, o dia a ser previsto correspondeu ao *cluster K2*, enquanto que os dias 4, 8 e 47 (“pré-selecionados”) corresponderam aos *clusters K1*, *K2* e *K2*, respectivamente. Desta forma, a potência de saída da usina foi prevista pela média da geração dos dias 8 e 47 (de acordo com a base de dados de potência - *PV data*), conforme representado na curva em azul.

Figura 14 – Exemplo de aplicação do algoritmo *Pattern Sequence-based Forecasting* 2 (PSF2)



Fonte: adaptado de: (WANG *et al.*, 2017).

Em resumo, o PSF2 é uma extensão do PSF1 no sentido de que utiliza a base de dados meteorológicos completa e faz uma segunda verificação por meio de clusterização, a partir

da base de dados reduzida e a previsão meteorológica para o dia a ser previsto. Analogamente aos métodos PSF e PSF1, o algoritmo contempla previsões para quaisquer horizontes de tempo, a partir da inclusão da previsão para o dia $d+1$ na base original e seguindo em *loop* com as demais previsões dia-a-dia, abrangendo o número de dias desejado.

O tamanho ótimo w da janela, como nos algoritmos anteriores, depende de cada implementação. Neste trabalho, a otimização do w foi novamente feita por meio de CV.

3.2 Aprendizagem supervisionada

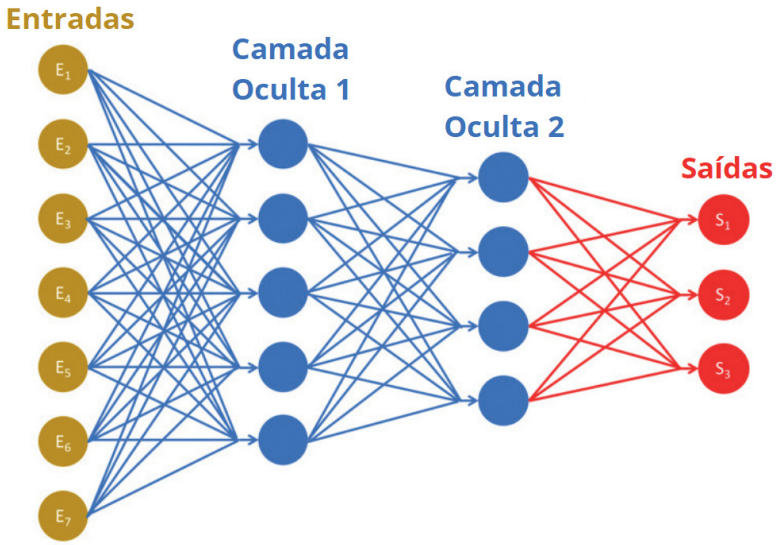
Aprendizagem supervisionada é uma categoria do Aprendizado de Máquina (*Machine Learning* – ML) na qual os modelos são apresentados a dados de treinamento associados às suas respectivas "repostas certas", seja um rótulo de classificação ou um valor numérico (regressão). Nesse tipo de aprendizagem, o modelo aprende as relações entre atributos e reposta corrigindo sua saída de acordo exatamente com a reposta certa (VANDERPLAS, 2017). Este trabalho fez uso dos seguintes modelos de aprendizagem supervisionada: Rede Neural Artificial e *Extreme Gradient Boosting* (*XGBoost*).

3.2.1 Rede Neural Artificial (RNA)

Dando sequência aos algoritmos implementados para a previsão da geração de energia, fez-se uso de um modelo de RNA, tendo em vista sua ampla utilização na literatura, conforme seção 1.2, bem como devido sua alta capacidade para resolução de problemas complexos e de natureza não-linear (NEGNEVITSKY, 2005). As RNAs são sistemas computacionais baseados nas redes neurais biológicas que compõem o cérebro humano. A partir de nós (neurônios) conectados entre si há a constituição de uma rede na qual sinais/dados de entrada são processados e propagados, por meio de sinapses, gerando sinais/dados de saída (OLIVEIRA, 2018). A Figura 15 apresenta a representação uma rede neural artificial com sete nós de entrada, duas camadas internas e três nós de saída.

O modelo de RNA adotado para implementação foi o Perceptron Multicamadas (*Multilayer Perceptron* – MLP) em razão deste apresentar bom desempenho em aplicações de regressão e alta capacidade de generalização, haja vista as habilidades de aprendizado e extração de características, em decorrência da construção em multicamadas (SILVA *et al.*, 2010). Além disso, conforme mencionado na seção 1.2 e reforçado por (SOBRI *et al.*, 2018) e (MELLIT;

Figura 15 – Representação de uma Rede Neural Artificial (RNA)



Fonte: adaptado de: (OLIVEIRA, 2018).

PAVAN, 2010), o modelo MLP é amplamente utilizado na literatura para aplicações de previsão de geração de energia elétrica proveniente de fonte solar.

3.2.2 Extreme Gradient Boosting (XGBoost)

Extreme Gradient Boosting (*XGBoost*) é um algoritmo baseado em árvores de decisão combinado a *Gradient Boosting* (aumento de gradiente), caracterizado por construir diversas árvores de decisão para extrair os atributos do conjunto de dados e, por meio de aprendizagem supervisionada, implementar o *Gradient Boosting* para favorecimento (impulso/*boosting*) dos parâmetros previsores mais proeminentes, de acordo com a correlação entre entrada e saída (GÉRON, 2019). O funcionamento geral do algoritmo se dá da seguinte forma: inicialmente, uma árvore de decisão é treinada no conjunto de dados; em seguida, uma segunda árvore é treinada nos resíduos da árvore anterior; o processo se repete até que a condição de parada seja atingida (usualmente um limiar de erro) e os resultados ponderados das árvores determinam a previsão (ANTONANZAS *et al.*, 2017).

A opção pela implementação do algoritmo *XGBoost* neste trabalho se deu por se tratar de um algoritmo em ascensão na literatura de previsão da geração de energia solar fotovoltaica, conforme (SOBRI *et al.*, 2018), bem como devido o algoritmo apresentar menor custo computacional frente a outros modelos de inteligência artificial, inclusos os demais algoritmos implementados neste trabalho, conforme (MACHADO *et al.*, 2021).

3.3 Modelos híbridos

Nesta seção são apresentados os modelos híbridos que foram implementados neste trabalho, sendo estes assim chamados por combinarem ao menos dois dos algoritmos abordados anteriormente em um modelo único, visando sobretudo buscar complementariedades entre os modelos, de modo que as previsões fossem otimizadas.

3.3.1 Pattern Sequence Neural Network

Os algoritmos de Rede Neural em Sequência de Padrões (*Pattern Sequence Neural Network* – PSNN) foram definidos em (LIN *et al.*, 2019) e consistem em modificações dos modelos *Pattern Sequence-based Forecasting*, abordados nas seções 3.1.1 a 3.1.3. Para os modelos PSNN, as etapas de clusterização e reconhecimento de sequências foram preservadas, porém, no lugar de realizar a previsão da geração do dia a ser previsto ($d+1$) pela média dos dias identificados pelo algoritmo PSF, utilizou-se estes dias "selecionados" para o treinamento de uma rede neural (do inglês, *Neural Network*), sendo a saída (*output*) da rede a própria previsão desejada.

Tendo em vista as razões citadas anteriormente na seção 3.2.1 e também visando uma comparação mais coerente entre os diferentes modelos implementados, foi escolhida uma rede neural MLP para cada um dos modelos, doravante denominados PSF-MLP, PSF1-MLP e PSF2-MLP.

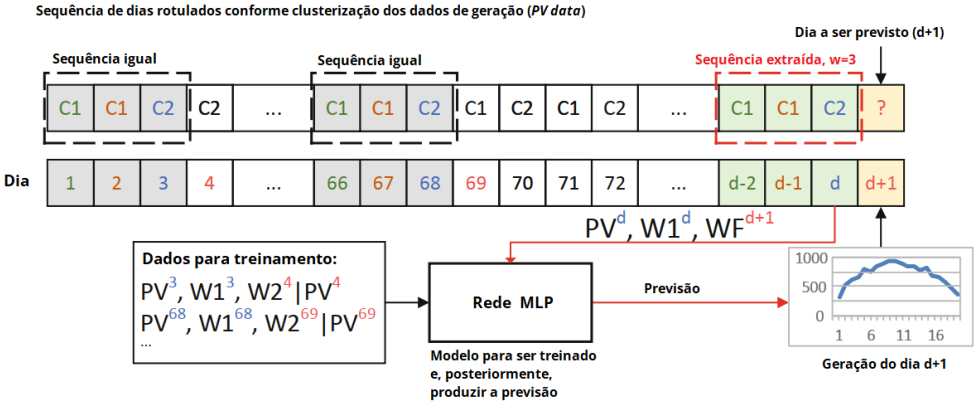
3.3.1.1 PSF-MLP

O algoritmo PSF-MLP consistiu em executar a clusterização e reconhecimentos de sequências conforme em 3.1.1 e com os dias "selecionados" treinar a rede MLP, utilizando como entrada (*input*) os dados de geração e meteorológicos (*PV data* e W1) dos respectivos dias anteriores e os dados meteorológicos (W2) do dia a ser previsto, associados à geração dos dias selecionados (*PV data*); a previsão para o dia $d+1$ foi então obtida executando o modelo, utilizando os dados de geração e meteorológicos do dia anterior (d) e de previsão meteorológica (WF) do dia a ser previsto ($d+1$).

A Figura 16 apresenta um esquemático do método PSF-MLP. No exemplo, os dados foram clusterizados em 2 *clusters*, de *labels C1* e *C2*; a janela tem tamanho w igual a 3 e, com isso, contém o padrão (*C1*, *C1*, *C2*). A busca na série temporal de rótulos localizou a sequência

nos dias 1 a 3 e 66 a 68. Assim, a rede MLP foi treinada com os dados dos dias 3, 4, 68 e 69, conforme supracitado e, então, a previsão da geração (potência de saída da usina) feita pela execução do modelo com os dados dos dias d e $d+1$.

Figura 16 – Exemplo de aplicação do algoritmo PSF-MLP



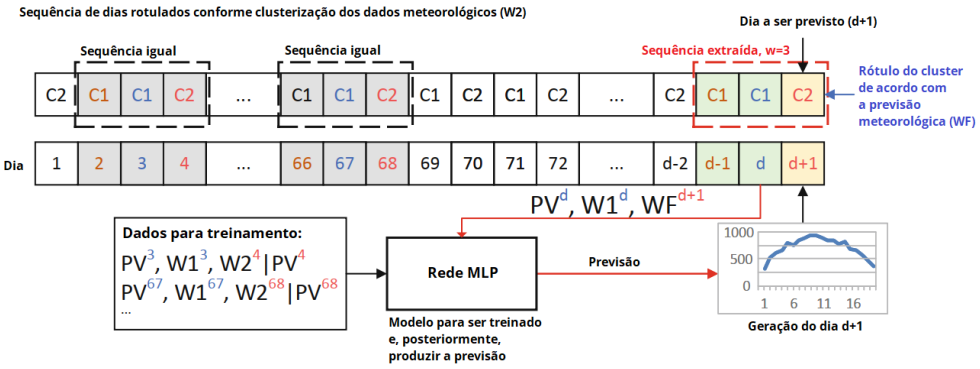
Fonte: adaptado de: (LIN *et al.*, 2019).

3.3.1.2 PSF1-MLP

O algoritmo PSF1-MLP consistiu em executar a clusterização e reconhecimentos de sequências conforme em 3.1.2 e com os dias "selecionados" treinar a rede MLP, com os dados estruturados conforme descrito na seção anterior (3.3.1.1)

A Figura 17 apresenta um esquemático do método PSF1-MLP. No exemplo, os dados foram clusterizados em 2 *clusters*, de *labels* *C1* e *C2*; a janela tem tamanho w igual a 3 e, com isso, contém o padrão (*C1*, *C1*, *C2*). A busca na série temporal de rótulos localizou a sequência nos dias 2 a 4 e 66 a 68. Assim, a rede MLP foi treinada com os dados dos dias 3, 4, 67 e 68, conforme supracitado e, então, a previsão da geração (potência de saída da usina) feita pela execução do modelo com os dados dos dias d e $d+1$.

Figura 17 – Exemplo de aplicação do algoritmo PSF1-MLP



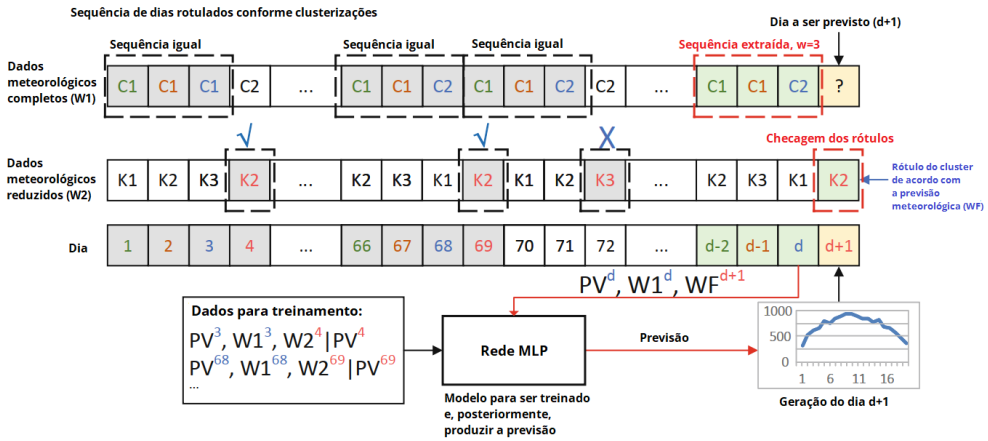
Fonte: adaptado de: (LIN *et al.*, 2019).

3.3.1.3 PSF2-MLP

O algoritmo PSF2-MLP consistiu em executar a clusterização e reconhecimentos de sequências conforme seção 3.1.3 e, com os dias "selecionados" treinar a rede MLP, com os dados estruturados como descrito na seção 3.3.1.1.

A Figura 18 apresenta um esquemático do método PSF2-MLP. No exemplo, os dados foram clusterizados, conforme base de dados W1, em 2 *clusters*, de *labels C1* e *C2*; a janela tem tamanho *w* igual a 3 e, com isso, contém o padrão (*C1*, *C1*, *C2*). A busca na série temporal localizou a sequência nos dias 1 a 3, 66 a 68 e 69 a 71, de modo que os dias "pré-selecionados" foram 4, 69 e 72. Conforme a clusterização pela base de dados W2, o dia a ser previsto correspondeu ao *cluster K2*, enquanto que os dias 4, 69 e 72 da série temporal corresponderam aos *clusters K2*, *K2* e *K3*, respectivamente. Assim, a rede MLP foi treinada com os dados dos dias 3, 4, 68 e 69, conforme supracitado e, então, a previsão da geração feita pela execução do modelo com os dados dos dias d e $d+1$.

Figura 18 – Exemplo de aplicação do algoritmo PSF2-MLP



Fonte: adaptado de: (LIN *et al.*, 2019).

3.3.2 Pattern Sequence Extreme Gradient Boosting

Os algoritmos de *Pattern Sequence Extreme Gradient Boosting* (PSXGB) são propostos pelo autor, explorando o conceito dos algoritmos PSNN (LIN *et al.*, 2019), porém substituindo a aplicação de uma rede neural por um modelo de *XGBoost*, visando usufruir dos benefícios discutidos em 3.2.2. Em termos práticos, os conceitos discutidos na seção 3.3.1 se preservam, aplicando-se apenas um modelo diferente no treinamento e previsão da geração. Consequentemente, os esquemáticos apresentados nas figuras 16, 17 e 18 são condizentes com os modelos doravante denominados PSF-*XGBoost*, PSF1-*XGBoost* e PSF2-*XGBoost*, a menos, claro, das substituições de rede MLP por modelo *XGBoost*.

3.4 Otimização de hiperparâmetros

Tendo em vista a implementação dos modelos abordados anteriormente, fez-se necessária a seleção dos hiperparâmetros ótimos dos modelos. Hiperparâmetros são parâmetros que devem ser configurados antes do modelo ser treinado (VANDERPLAS, 2017), como o número de camadas da rede neural ou a taxa de aprendizado do modelo, por exemplo. Os hiperparâmetros ótimos dizem respeito à combinação, dentre as diferentes possibilidades, que resulta no modelo com maior acurácia (menor erro). Para determinação dos hiperparâmetros dos métodos de aprendizagem supervisionada (RNA e *XGBoost*), foram implementados três métodos de busca: *Grid Search*, *Randomized Search* e *Bayesian Optmization*, enquanto que para otimização dos tamanhos das janelas (w) dos métodos *Pattern Sequence-based Forecast* utilizou-se validação cruzada, conforme seções a seguir.

3.4.1 Grid Search

A *Grid Search* é uma técnica de ajuste de hiperparâmetros que consiste numa busca exaustiva pela combinação ótima de hiperparâmetros, dentro de um espaço de combinações possíveis. O método consiste, basicamente, em avaliar cada configuração possível do modelo, conforme os valores possíveis pré-determinados para os hiperparâmetros, e selecionar a combinação que apresentar menor erro (MALIK, 2020).

Justifica-se a utilização da *Grid Search* neste trabalho pelo fato que esta é uma das técnicas mais utilizadas para seleção de hiperparâmetros, além de que garante que todas as combinações pré-definidas serão avaliadas e que o modelo selecionado certamente será o melhor

dentro do conjunto de possibilidades (COLLINS, 2021). Apesar disto, exatamente por testar cada uma das possibilidades, o número de modelos avaliados cresce exponencialmente com a quantidade de hiperparâmetros considerada na busca, de modo que o custo computacional pode tornar-se elevado (GÉRON, 2019). Assim, também foram implementadas as técnicas de *Randomized Search* e *Bayesian Optmization*.

3.4.2 Randomized Search

A *Randomized Search* executa uma pesquisa aleatória, que consiste em selecionar, em cada iteração, uma combinação aleatória dos hiperparâmetros, avaliando uma quantidade de combinações determinada pelo número total de iterações (GÉRON, 2019). A principal vantagem dessa abordagem é que não é executada uma busca exaustiva, como na *Grid Search*, de modo que o custo computacional não é definido pelas combinações de hiperparâmetros, mas pelo número de iterações, de modo que o método tende a ser mais eficiente que a *Grid Search* (BERGSTRA; BENGIO, 2012). A desvantagem da técnica de *Randomized Search* é que todas as iterações ainda são independentes entre si, de modo que não há aprendizado de uma iteração para outra, o que faz com que sempre seja executado o número de iterações pré-definido (GUPTA, 2020), o que pode ainda ser computacionalmente ineficiente, a depender da aplicação. Nesse sentido, foi proposta também a utilização do método de *Bayesian Optmization*, descrito a seguir.

3.4.3 Bayesian Optmization

Bayesian Optmization é um algoritmo de Otimização Baseada em Modelo Sequencial (*Sequential Model-Based Optimization* – SBMO) que constrói um modelo probabilístico associando cada hiperparâmetro a uma probabilidade com respeito à função objetivo. Desse modo, o algoritmo utiliza os resultados das iterações anteriores para determinar os valores dos hiperparâmetros da próxima iteração (BROCHU *et al.*, 2010). Com essa abordagem, o algoritmo não faz buscas exaustivas, mas seleciona os hiperparâmetros conforme vão apresentando melhor desempenho. Em termos mais simples, o algoritmo "aprende o caminho" da solução ótima, por meio das funções de probabilidade, o que, em geral, proporciona resultados melhores, sobretudo do ponto de vista de eficiência computacional (SNOEK *et al.*, 2012).

Conforme implementação desenvolvida (capítulo 4), os três métodos de seleção de hiperparâmetros foram empregados, servindo como validação mútua, tendo em vista apresentarem resultados similares; porém, no que tange aos tempos de execução (esforço computacional),

a abordagem de *Bayesian Optmization* mostrou-se a mais eficiente.

3.4.4 Validação Cruzada

Validação Cruzada (CV) consiste no particionamento de um conjunto de dados em n subconjuntos de modo que seja possível treinar o modelo em determinados subconjuntos e validá-lo no subconjunto complementar dos dados (KOHAVI, 1995). Para seleção do tamanho ótimo das janelas dos métodos PSF e derivados, a validação cruzada foi feita considerando n igual a 12, tendo em vista que há 12 meses em um ano e a base de dados de treinamento abrange exatamente esse período, conforme capítulo 2. Assim, com os dados subdivididos conforme os meses do ano, procedeu-se com o treinamento em $n-1$ meses e validação em 1 mês, repetindo esse processo n vezes para diferentes valores de w. O w correspondente ao menor erro médio da validação cruzada é então determinado como w ótimo. A Figura 19 apresenta um esquemático simplificado da validação cruzada em questão.

Figura 19 – Esquemático simplificado da validação cruzada

Fonte: o próprio autor.

4 IMPLEMENTAÇÃO

Este capítulo descreve as implementações dos algoritmos de previsão da geração fotovoltaica, conforme foram introduzidos no capítulo 3. Todos os algoritmos foram implementados com treinamento a partir do primeiro ano dos dados (junho 2019 a junho de 2020) e com horizonte de previsão de 24h à frente para 365 dias consecutivos, de modo que pudessem ser feitos os testes de acurácia das previsões utilizando os dados restantes.

4.1 Modelos PSF

Os primeiros algoritmos de previsão implementados foram os modelos de *Pattern Sequence-based Forecast*, que foram descritos na seção 3.1. As seções a seguir detalham os passos seguidos em cada implementação.

4.1.1 PSF

Para a implementação do método PSF, fez-se uso do algoritmo desenvolvido por (SHENDE; BOKDE, 2019), com algumas adaptações feitas pelo autor para melhor conformidade com este trabalho, sobretudo no que diz repeito à validação cruzada para determinação do valor ótimo da janela (w) e nas dimensionalidades (*shapes*) dos conjuntos utilizados.

A otimização do tamanho da janela (w) para o método PSF foi feita de acordo com a metodologia detalhada na seção 3.4.4 e retornou como valor ótimo o número 3, valor este que foi adotado para implementação do algoritmo. O método recebeu como entradas os dias clusterizados pelos dados de geração (*PV data*), os próprios dados de geração (para posterior cálculo da previsão), o tamanho da janela (w), o número de *clusters* (K igual a 4) e a granularidade dos dados (12 dados por dia, haja vista 12 horas). Sua saída foi uma matriz 365×12, com as previsões horárias de potência de saída da usina, entre 05:00 e 17:00, para 365 dias, conforme supracitado.

4.1.2 PSF1

A implementação do método PSF1 tomou como base o trabalho de (SHENDE; BOKDE, 2019), feitas as modificações necessárias para que fosse executada a metodologia do PSF1, conforme seção 3.1.2.

A otimização do tamanho da janela (w) para o método PSF1 também foi feita de acordo com a metodologia detalhada na seção 3.4.4 e, novamente, retornou como valor ótimo o número 3, valor este que foi adotado para implementação do algoritmo. O método recebeu como entradas o número de *clusters* (K igual a 3), os dias clusterizados pelos dados meteorológicos reduzidos (*W2*), o tamanho da janela (w), a granularidade dos dados de geração (12 dados por dia, haja vista 12 horas) e os dados de geração (para posterior cálculo da previsão). Sua saída foi uma matriz 365×12, com as previsões horárias de potência de saída da usina, entre 05:00 e 17:00, para 365 dias, conforme supracitado.

4.1.3 PSF2

Para a implementação do método PSF2 também foi tomada como referência principal o trabalho de (SHENDE; BOKDE, 2019). A partir das funções e lógicas do mesmo, o autor desenvolveu as modificações necessárias para que fosse executada a metodologia do PSF2, conforme seção 3.1.3.

A otimização do tamanho da janela (w) para o método PSF2, também feita de acordo com a metodologia detalhada na seção 3.4.4, retornou como valor ótimo o número 3, sendo este adotado para implementação do algoritmo. O método recebeu como entradas os números de *clusters* das clusterizações pelos dados meteorológicos reduzidos (*W2*) e pelos dados meteorológicos completos (*W1*) – K igual a 3 e K igual a 6, respectivamente –, as duas respectivas clusterizações (pelas duas bases de dados), o tamanho da janela (w), a granularidade dos dados de geração (12 dados por dia, haja vista 12 horas) e os dados de geração (para posterior cálculo da previsão). Sua saída foi uma matriz 365×12, com as previsões horárias de potência de saída da usina, entre 05:00 e 17:00, para 365 dias, conforme supracitado.

4.2 Modelos RNA

Conforme abordado na seção 3.2.1, o modelo de RNA adotado para implementação foi uma rede MLP. Este trabalho implementou dois modelos da referida rede, um treinado apenas com dados de geração (*PV data*) e outro treinado também com dados meteorológicos da usina. Isso foi feito tendo em vista que é possível que os dados meteorológicos nem sempre sejam de fácil aquisição, de modo que pareceu relevante a implementação – e avaliação – de um algoritmo que utilizasse apenas dados elétricos (potência de geração). Ainda assim, tendo em

vista o entendimento de que um modelo mais robusto, utilizando também dados meteorológicos, tenderia a apresentar melhor desempenho, seguiu-se com a implementação de um segundo modelo, com essa característica.

As opções de hiperparâmetros para seleção foram: função de ativação: tangente hiperbólica (*tanh*) e Unidade Linear Retificada (*Rectified Linear Unit* – ReLU); otimizador: Adam e Gradiente Descendente Estocástico (*Stochastic Gradient Descent* – SGD); taxa de aprendizado: constante e adaptativa; regularização L2: 0,0001; 0,001; 0,01 e 0,05; tamanhos das camadas ocultas: uma camada de 20 neurônios, duas camadas de 10 neurônios e três camadas de tamanhos 5, 10 e 5, respectivamente.

4.2.1 MLP1

O primeiro modelo de rede MLP que foi implementado, que será denominado MLP1 daqui em diante, foi treinado utilizando apenas os dados de geração de energia da usina, sendo utilizada a classe *MLPRegressor* da biblioteca Python *Scikit-learn*.

Conforme mencionado na seção 3.4, foram adotados três métodos para otimização dos hiperparâmetros da rede, sendo selecionados aqueles com maior ocorrência. A Tabela 4 apresenta os valores ótimos segundo cada método de seleção, com destaque (negrito) para os valores selecionados para a configuração da rede MLP1.

Tabela 4 – Melhores hiperparâmetros para o modelo MLP1

Hiperparâmetro	*Grid Search*	*Randomized Search*	*Bayesian Optmization*
Função de ativação	***tanh***	***tanh***	***tanh***
Otimizador	**Adam**	**Adam**	**Adam**
Taxa de aprendizado (tipo)	**Constante**	**Constante**	Adaptativa
Regularização L2	**0,05**	**0,05**	0,001
Tamanhos das camadas ocultas	**(10, 10)**	**(10, 10)**	**(10, 10)**

Fonte: o próprio autor.

Assim, o modelo foi configurado com função de ativação *tanh*, otimizador *Adam*, taxa de aprendizado constante, regularização L2 igual a $0,05$ e duas camadas ocultas de tamanho 10. Foi feito o treinamento com a base de dados de geração (conforme seção 2.1) e feita a previsão das próximas 24h por 365 dias consecutivos.

4.2.2 MLP2

O segundo modelo de rede MLP que foi implementado, que será denominado MLP2 daqui em diante, foi treinado utilizando todos os dados disponíveis, tanto de geração como meteorológicos. Novamente fez-se uso da classe *MLPRegressor* da biblioteca Python *Scikit--learn*.

A otimização dos hiperparâmetros da rede, assim como no modelo anterior, foi feita a partir da seleção dos hiperparâmetros mais recorrentes dentre os três métodos aplicados. A Tabela 5 apresenta os valores ótimos segundo cada método de seleção, com destaque (negrito) para os valores selecionados para a configuração da rede MLP2.

Tabela 5 – Melhores hiperparâmetros para o modelo MLP2

Hiperparâmetro	*Grid Search*	*Randomized Search*	*Bayesian Optmization*
Função de ativação	**ReLU**	**ReLU**	**ReLU**
Otimizador	**SGD**	**SGD**	**SGD**
Taxa de aprendizado (tipo)	**Adaptativa**	**Adaptativa**	**Adaptativa**
Regularização L2	**0,05**	**0,05**	0,001
Tamanhos das camadas ocultas	**(5, 10, 5)**	**(5, 10, 5)**	(10, 10)

Fonte: o próprio autor.

Assim, o modelo foi configurado com função de ativação ReLU, otimizador SGD, taxa de aprendizado adaptativa, regularização L2 igual a $0,05$ e três camadas ocultas de tamanhos 5, 10, e 5, respectivamente. Foi feito o treinamento com as bases de dados de geração e meteorológica (conforme seção 2.1) e feita a previsão para 365 dias futuros.

4.3 Modelos XGBoost

Este trabalho implementou dois modelos de *XGBoost*, um treinado apenas com dados de geração (*PV data*) e outro treinado também com dados meteorológicos da usina. Isso foi feito pelas mesmas razões citadas anteriormente na implementação dos modelos MLP (seção 4.2), bem como visando replicar o que foi adotado para a RNA, de modo a permitir melhor comparação entre os algoritmos.

A otimização dos hiperparâmetros dos modelos *XGBoost* também foi realizada conforme anteriormente, em que foram verificados três métodos (*Grid Search*, *Randomized Search* e *Bayesian Optmization*) e selecionados aqueles com maior ocorrência. As opções de hiperparâmetros para seleção foram: peso mínimo (*min_child_weight*): 1, 5 e 10; pseudo-

-regularização (*gamma*): 0,5, 1, 1,5, 2 e 5; subamostra (*subsample*): 0,6, 0,8 e 1; profundidade máxima por árvore (*max_depth*): 3, 4 e 5; taxa de aprendizado (*learning_rate*): 0,01, 0,02 e 0,03; regularização L1 (*reg_alpha*): 0, 10, 20 e 30; fração de colunas amostradas (*colsample_bytree*): 0,6, 0,8 e 1.

4.3.1 XGBoost1

O primeiro modelo *XGBoost* que foi implementado, que será denominado *XGBoost1* daqui em diante, foi treinado utilizando apenas os dados de geração de energia da usina, sendo utilizada o pacote Python *xgboost*.

Conforme mencionado na seção 3.4, foram adotados três métodos para otimização dos hiperparâmetros da rede, sendo selecionados aqueles com maior ocorrência. A Tabela 6 apresenta os valores ótimos segundo cada método de seleção, com destaque (negrito) para os valores selecionados para a configuração da rede *XGBoost1*.

Tabela 6 – Melhores hiperparâmetros para o modelo *XGBoost1*

Hiperparâmetro	*Grid Search*	*Randomized Search*	*Bayesian Optmization*
min_child_weight	**1**	**1**	5
gamma	0,5	**1**	**1**
subsample	**1**	**1**	**1**
max_depth	3	**5**	**5**
learning_rate	0,01	**0,03**	**0,03**
reg_alpha	**10**	**10**	**10**
colsample_bytree	**1**	**1**	**1**

Fonte: o próprio autor.

Assim, o modelo foi configurado com peso mínimo, pseudo-regularização e subamostra todas iguais a 1, profundidade máxima por árvore igual a 5, taxa de aprendizado igual a 0,03, regularização L1 igual a 10 e fração de colunas amostradas igual a 1.

4.3.2 XGBoost2

O segundo modelo *XGBoost* que foi implementado, que será denominado *XGBoost2* daqui em diante, foi treinado utilizando todos os dados disponíveis, tanto de geração como meteorológicos. Novamente fez-se uso do pacote *xgboost*, agora com apoio da classe *MultiOutputRegressor* para aplicação *multi-output*, esta última proveniente da biblioteca Python *Scikit-learn*.

A otimização dos hiperparâmetros do modelo, assim como no caso anterior, foi feita

a partir da seleção dos hiperparâmetros mais recorrentes dentre os três métodos aplicados. A Tabela 7 apresenta os valores ótimos segundo cada método de seleção, com destaque (negrito) para os valores selecionados para a configuração da rede *XGBoost2*.

Tabela 7 – Melhores hiperparâmetros para o modelo *XGBoost2*

Hiperparâmetro	*Grid Search*	*Randomized Search*	*Bayesian Optmization*
min_child_weight	**5**	**5**	**5**
gamma	**2**	**2**	**2**
subsample	**0,6**	**0,6**	**0,6**
max_depth	3	**4**	**4**
learning_rate	**0,03**	**0,03**	**0,03**
reg_alpha	**0**	**0**	**0**
colsample_bytree	**1**	**1**	**1**

Fonte: o próprio autor.

Assim, o modelo foi configurado com peso mínimo igual a 5, pseudo-regularização igual a 2, subamostra igual a 0,6, profundidade máxima por árvore igual a 4, taxa de aprendizado igual a 0,03, regularização L1 igual a 0 e fração de colunas amostradas igual a 1.

4.4 Modelos híbridos

Conforme discutido na seção 3.3, este trabalho implementou seis métodos híbridos para previsão da geração de energia elétrica da usina, todos fazendo uso de uma abordagem de *Pattern Sequence-based Forecast* associada a uma rede neural ou um modelo *XGBoost*.

As otimizações dos hiperparâmetros do modelos PSF, abordadas nas seções anteriores, foram preservadas, haja vista que a 1ª parte dos algoritmos híbridos consiste na mesma execução do respectivo PSF (PSF, PSF1 ou PSF2), enquanto que não houve otimização de hiperparâmetros específica para os modelos de MLP e *XGBoost* utilizados, em razão de que a cada dia de previsão é treinado um novo modelo, específico para a previsão daquele dia, de modo que cada modelo híbrido faz o treinamento de 365 modelos de RNA/*XGBoost*. Caso fossem feitas otimizações para cada modelo e cada algoritmo híbrido, o custo computacional seria impraticável. Assim, preservou-se as configurações selecionadas anteriormente e os modelos diferenciaram-se entre si pelo treinamento.

4.4.1 PSF-MLP

Para a implementação do método PSF-MLP, novamente, fizeram-se adaptações no trabalho de (SHENDE; BOKDE, 2019). Em relação ao algoritmo que implementa o modelo

PSF, a distinção se dá que não é mais calculada uma média aritmética como valor da previsão, mas um modelo MLP é treinado, conforme descrito na seção 3.3.1.1 e a saída da rede determina a previsão.

O método recebeu como entradas os dias clusterizados pelos dados de geração (*PV data*), o tamanho da janela (*w*), o número de *clusters* (K igual a 4), a granularidade dos dados (12 dados por dia, haja vista 12 horas) e as três bases de dados (*PV data*, *W1* e *W2*). Sua saída foi uma matriz 365×12, com as previsões horárias de potência de saída da usina, entre 05:00 e 17:00, para 365 dias, de igual modo aos demais modelos.

4.4.2 PSF-XGBoost

Assim como para a implementação do método PSF-MLP, novamente fez-se adaptações no trabalho de (SHENDE; BOKDE, 2019) para desenvolvimento do método *PSF-XGBoost*. Em relação ao algoritmo que implementa o modelo PSF-MLP, a diferença fica por conta de que, para previsão, se utiliza um modelo de *XGBoost*, treinado nos dados selecionados pelo PSF, conforme descrito na seção 3.3.2.

O método recebeu as mesmas entradas do algoritmo PSF-MLP, retornando uma saída de características também iguais.

4.4.3 PSF1-MLP

Para a implementação do método PSF1-MLP, uma vez mais, fez-se adaptações no trabalho de (SHENDE; BOKDE, 2019). Em comparação ao algoritmo que implementa o modelo PSF1, a distinção se dá que não é mais calculada uma média aritmética como valor da previsão, mas um modelo MLP é treinado, conforme descrito na seção 3.3.1.2 e a saída da rede determina a previsão.

O método recebeu como entradas os dias clusterizados pelos dados meteorológicos reduzidos (*W2*), o tamanho da janela (*w*), o número de *clusters* (K igual a 3), a granularidade dos dados de geração (12 dados por dia, haja vista 12 horas) e as três bases de dados (*PV data*, *W1* e *W2*). Sua saída foi uma matriz 365×12, com as previsões horárias de potência de saída da usina, entre 05:00 e 17:00, para 365 dias, de igual modo aos demais modelos.

4.4.4 PSF1-XGBoost

De igual modo à implementação do método PSF1-MLP, foram feitas adaptações no trabalho de (SHENDE; BOKDE, 2019) para desenvolvimento do método *PSF1-XGBoost*. Em relação ao algoritmo que implementa o modelo PSF1-MLP, a diferença se dá na utilização para previsão de um modelo *XGBoost*, treinado nos dados selecionados pelo PSF, conforme descrito na seção 3.3.2.

O método recebeu as mesmas entradas do algoritmo PSF1-MLP, retornando uma saída de características também iguais, como todos os demais métodos deste trabalho.

4.4.5 PSF2-MLP

O método PSF2-MLP, assim como os demais métodos híbridos já abordados, também foi implementado a partir de adaptações do trabalho de (SHENDE; BOKDE, 2019). Em relação ao algoritmo que implementa o modelo PSF2, a diferença se dá em que a previsão não é mais calculada por meio de uma média aritmética, mas por um um modelo MLP, treinado conforme descrito na seção 3.3.1.3.

O método recebeu como entradas os dias clusterizados pelos dados meteorológicos reduzidos (*W2*) e pelos dados meteorológicos completos (*W1*), o tamanho da janela (w), os números de *clusters* de cada clusterização (K igual a 3 e K igual a 6, respectivamente), a granularidade dos dados de geração (12 dados por dia, haja vista 12 horas) e as três bases de dados (*PV data*, *W1* e *W2*). Sua saída foi uma matriz 365×12, com as previsões horárias de potência de saída da usina, entre 05:00 e 17:00, para 365 dias, semelhantemente aos modelos anteriores.

4.4.6 PSF2-XGBoost

Finalmente, aborda-se a implementação do algoritmo *PSF2-XGBoost* no qual, assim como os demais algoritmos híbridos, foram feitas adaptações no trabalho de (SHENDE; BOKDE, 2019) para desenvolvimento do método. Em relação ao algoritmo que implementa o modelo PSF2-MLP, a diferença se dá na utilização para previsão de um modelo *XGBoost*, treinado nos dados selecionados pelo PSF, conforme descrito na seção 3.3.2.

O método recebeu as mesmas entradas do algoritmo PSF2-MLP, retornando uma saída de características também iguais, assim como os demais métodos deste trabalho.

5 RESULTADOS E DISCUSSÃO

Para avaliação dos resultados dos algoritmos implementados, utilizou-se como métricas de desempenho a Raiz do Erro Quadrático Médio (*Root Mean Squared Error* – RMSE) e o Erro Absoluto Médio (*Mean Absolute Error* – MAE), tendo em vista serem os indicadores mais utilizados na literatura (BOTCHKAREV, 2019). Seus cálculos foram feitos a partir das funções pré-existentes na biblioteca *Scikit-learn*.

Como *baseline* de comparação do métodos, foi utilizado um modelo de previsão de persistência no qual a geração do dia $d+1$ é prevista como igual à do dia anterior (d), em semelhança a (LIN *et al.*, 2019). A Tabela 8 apresenta os indicadores supracitados para todos os modelos de previsão implementados, bem como modelo de persistência.

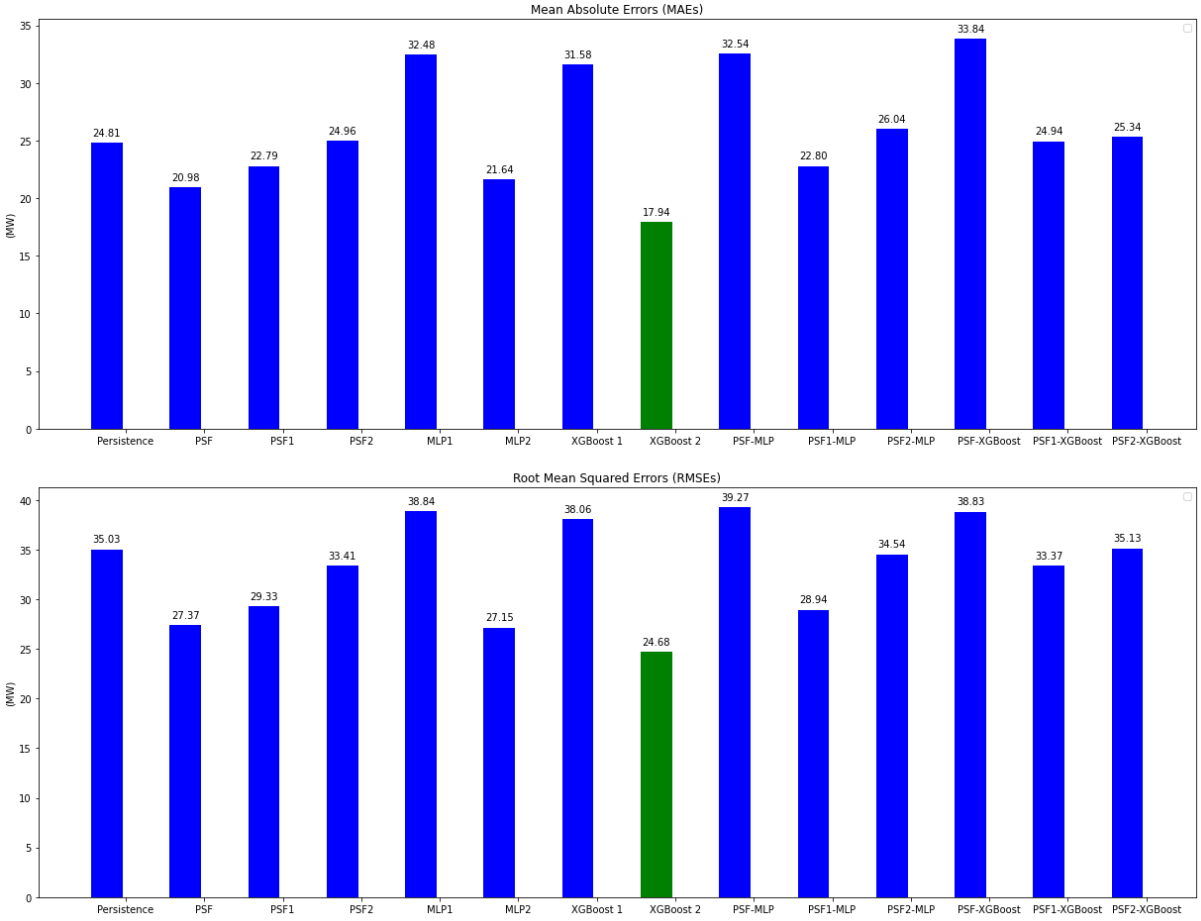
Tabela 8 – Desempenho dos modelos de previsão implementados

Modelo	MAE (MW)	RMSE (MW)
Persistência	24,37	34,34
PSF	20,59	26,75
PSF1	22,02	28,20
PSF2	29,13	38,76
MLP1	32,20	38,67
MLP2	21,09	26,40
XGBoost1	31,23	37,73
XGBoost2	**17,12**	**23,50**
PSF-MLP	32,33	38,82
PSF-XGBoost	33,72	38,69
PSF1-MLP	22,00	27,68
PSF1-XGBoost	22,53	30,20
PSF2-MLP	32,12	42,21
PSF2-XGBoost	32,07	42,52

Fonte: o próprio autor.

Objetivando uma melhor visualização da comparação de desempenho entre os modelos, a Figura 20 apresenta um gráfico de barras com os valores de MAE e RMSE obtidos.

Figura 20 – Desempenho dos modelos de previsão implementados



Fonte: o próprio autor.

Como principais resultados, verificou-se que:

– O melhor modelo de previsão foi *XGBoost*2, que foi treinado com todos os dados disponíveis – tanto de geração quanto meteorológicos –, tendo apresentado o melhor desempenho conforme ambos os indicadores, mostrando o grande potencial do algoritmo *XGBoost* para o tema do trabalho.

– O segundo melhor modelo de previsão foi o MLP2, que assim como o *XGBoost*2 foi treinado com todos os dados disponíveis, o que aponta tanto para o potencial das redes neurais para a aplicação aqui prevista – fato já observado na literatura –, como também para o impacto positivo dos treinamentos utilizando dados meteorológicos, de modo que a correlação entre condições climáticas e geração de energia em usinas solares se mostrou bastante evidente nos melhores modelos.

– O terceiro melhor modelo foi o PSF, que apesar de não ter reproduzido todo o potencial identificado quando inicialmente proposto, mostrou-se bastante competitivo frente aos demais modelos, sobretudo quando considerados apenas aqueles que não

utilizaram dados meteorológicos (entre estes, foi o melhor). Com isso, em situações em que estes dados não estejam disponíveis, o algoritmo PSF apresenta-se como uma boa opção para previsão da geração de energia da planta.

– O melhor modelo híbrido foi o PSF1-MLP, resultado condizente com (LIN *et al.*, 2019), o que indica certa vantagem em se utilizar dados meteorológicos na etapa de clusterização. No entanto, a adição de uma segunda etapa de clusterização (PSF2 e modelos híbridos baseados neste) não mostrou ganhos neste trabalho.

Adicionalmente, foram observados também os tempos de execução de cada modelo implementado, visando quantificar o custo computacional de cada método. A Tabela 9 apresenta os tempos de execução demandados por cada modelo, tanto nas etapas auxiliares (otimização de hiperparâmetros e clusterizações), como na execução do algoritmo de previsão propriamente dito. É possível observar que, novamente, os modelos de RNA (MLP) e *XGBoost* apresentaram os melhores desempenhos e que, apesar de em valores proporcionais existirem diferenças significativas (modelo mais demorado demandou cerca de 13,5 vezes o tempo do modelo mais rápido, por exemplo), em termos de valores absolutos, todos os modelos tiveram tempos de execução, aproximadamente, até 150 segundos (2,5 minutos), de modo que todos os modelos podem ser considerados computacionalmente eficientes para o propósito da aplicação. Todas as execuções foram realizadas no ambiente virtual *Google Colab* (versão gratuita).

Tabela 9 – Custo computacional (tempos de execução) dos modelos de previsão implementados

Modelo	Tempo de execução (s)		
	Etapas auxiliares	Execução do modelo	Total
PSF	36,629	9,194	45,823
PSF1	37,675	7,689	45,364
PSF2	38,029	16,662	54,691
MLP1	26,659	62,904	89,563
MLP2	10,599	0,459	11,058
XGBoost1	16,221	52,774	68,995
XGBoost2	44,801	0,298	45,099
PSF-MLP	47,228	35,488	82,716
PSF-XGBoost	81,430	15,906	97,336
PSF1-MLP	48,274	27,037	75,311
PSF1-XGBoost	82,476	56,868	139,344
PSF2-MLP	48,628	43,389	92,017
PSF2-XGBoost	82,830	67,212	150,042

Fonte: o próprio autor.

6 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, foi avaliada a aplicação de métodos de Aprendizagem de Máquina para a previsão da geração de energia elétrica de uma usina solar fotovoltaica, visando beneficiar a integração e operação de usinas dessa natureza frente à característica de variação da potência de saída. Para tal, foram implementados diferentes modelos, com diferentes metodologias, notadamente métodos baseados em reconhecimento de sequências, redes neurais artificiais (RNAs), florestas aleatórias (o algoritmo *XGBoost*) e métodos híbridos. Os modelos receberam como entradas os dados históricos de geração da usina e, em alguns modelos, também a série histórica de dados meteorológicos.

Conforme resultados obtidos, verificou-se que o algoritmo *XGBoost* – treinado com todos os dados disponíveis – apresentou o melhor desempenho para previsão, seguido pelo modelo MLP, também treinado com dados elétricos (geração de energia) e meteorológicos. Tais resultados apontaram para a eficiência de métodos de Inteligência Artificial (IA) para a aplicação, bem como para a relevância da utilização de dados relativos às condições climáticas na usina para a execução da previsão da geração. A abordagem de *Pattern Sequence-based Forecast* (PSF), que é mais transparente do que uma RNA ou mesmo o *XGBoost* – o que, em geral, pode ser algo vantajoso –, mostrou-se competitiva, tendo o melhor desempenho dentre os modelos treinados apenas com dados históricos de geração de energia elétrica.

Para trabalhos futuros, pode-se estender a avaliação para métodos proeminentes na literatura e que não foram contemplados neste trabalho, como Máquina de Vetores de Suporte (SVM), Redes Neurais Convolucionais (RNCs) e *Adaptive Boosting* (*AdaBoost*). Sugere-se também a utilização de dados de outra natureza – tais como sujidade, taxa de desempenho (*performance ratio*), ocorrências de manutenção no sistema, entre outros – como dados complementares nos modelos de previsão, avaliando se estes acrescentam informações relevantes aos algoritmos (numa espécie de "ajuste fino", com informações mais minuciosas). Finalmente, é sugerida a experimentação de outras abordagens para combinação de métodos distintos, de maneira diferente à construção dos modelos híbridos deste trabalho; pode-se aplicar uma abordagem *emsemble*, combinando os métodos por *blending* ou *stacking*, por exemplo.

REFERÊNCIAS

AGGARWAL, C.; REDDY, C. **Data Clustering: Algorithms and Applications**. Boca Raton, FL: Chapman and Hall/CRC, 2014. v. 1. 89-90 p.

AHMED, R.; SREERAM, V.; MISHRA, Y.; ARIF, M. D. A review and evaluation of the state-of-the-art in pv solar power forecasting: Techniques and optimization. **Renewable and Sustainable Energy Reviews**, v. 124, p. 109792, maio 2020.

ANTONANZAS, J.; OSORIO, N.; ESCOBAR, R.; URRACA, R.; MARTÍNEZ-DE-PISÓN, F. J.; ANTONANZAS-TORRES, F. Review of photovoltaic power forecasting. **Solar Energy**, Elsevier, v. 136, p. 78–111, 10 2016.

ANTONANZAS, J.; URRACA, R.; PERNÍA-ESPINOZA, A.; ALDAMA, A.; FERNÁNDEZ-JIMÉNEZ, L. A.; MARTÍNEZ-DE-PISÓN, F. J. Single and Blended Models for Day-Ahead Photovoltaic Power Forecasting. In: **Hybrid Artificial Intelligent Systems**. Cham: Springer International Publishing, 2017. (Lecture Notes in Computer Science), p. 427–434. ISBN 9783319596501.

ARAIN, F. N. **Decision Tree Classification Algorithm**. 2021? Disponível em: <https://www.devops.ae/decision-tree-classification-algorithm/>. Acesso em: 20 nov. 2021.

BENZAKE, Y. **Tout ce que vous voulez savoir sur l'algorithme K-Means**. 2018. Disponível em: <https://mrmint.fr/algorithme-k-means>. Acesso em: 14 ago. 2021.

BERGSTRA, J.; BENGIO, Y. Random search for hyper-parameter optimization. **Journal of Machine Learning Research**, v. 13, n. 10, p. 281–305, 2012. Disponível em: <http://jmlr.org/papers/v13/bergstra12a.html>.

BOTCHKAREV, A. Performance metrics (error measures) in machine learning regression, forecasting and prognostics: Properties and typology. **Interdisciplinary Journal of Information, Knowledge, and Management**, v. 14, p. 45–79, 2019.

BP. BP Statistical Review of World Energy 2021. v. 70, 2021. Disponível em: <https://www.bp.com/en/global/corporate/energy-economics/statistical-review-of-world-energy.html>.

BROCHU, E.; CORA, V. M.; FREITAS, N. de. A tutorial on bayesian optimization of expensive cost functions, with application to active user modeling and hierarchical reinforcement learning. **arXiv:1012.2599 [cs]**, 12 dez. 2010. Disponível em: <http://arxiv.org/abs/1012.2599>.

BROWNLEE, J. **Time Series Forecasting as Supervised Learning**. 2016. Disponível em: <https://machinelearningmastery.com/time-series-forecasting-supervised-learning/>. Acesso em: 21 set. 2021.

BROWNLEE, J. **A Gentle Introduction to Normality Tests in Python**. 2019. Disponível em: <https://machinelearningmastery.com/a-gentle-introduction-to-normality-tests-in-python/>. Acesso em: 15 nov. 2021.

CHU, Y.; LI, M.; COIMBRA, C. F.; FENG, D.; WANG, H. Intra-hour irradiance forecasting techniques for solar power integration: a review. **iScience**, v. 24, n. 10, p. 103136, set. 2021.

COLLINS, A. **Using Grid Search to Optimize Hyperparameters**. 2021. Disponível em: <https://www.section.io/engineering-education/grid-search/>. Acesso em: 29 ago. 2021.

DAS, K. R.; IMON, A. H. M. R. A brief review of tests for normality. **American Journal of Theoretical and Applied Statistics**, v. 5(1), p. 5–12, jan. 2016.

DIAGNE, M.; DAVID, M.; LAURET, P.; BOLAND, J.; SCHMUTZ, N. Review of solar irradiance forecasting methods and a proposition for small-scale insular grids. **Renewable and Sustainable Energy Reviews**, v. 27, p. 65–76, nov. 2013.

EBERLY COLLEGE OF SCIENCE. **Influential Points**. Pennsylvania: [*S. n.*], 2021? Disponível em: <https://online.stat.psu.edu/stat501/lesson/11/>. Acesso em: 25 dez. 2021.

ENEL GREEN POWER. **Parque solar São Gonçalo**. 2021. Disponível em: <https://www.enelgreenpower.com/pt/nossos-projetos/highlights/parque-solar-sao-goncalo/>. Acesso em: 21 jan. 2022.

EPE. Balanço Energético Nacional 2021: Ano base 2020 / Empresa de Pesquisa Energética. Rio de Janeiro, 2021. Disponível em: <https://www.epe.gov.br/pt/publicacoes-dados-abertos/publicacoes/balanco-energetico-nacional-2021>.

GAMA, J.; MEDAS, P.; RODRIGUES, P. Concept drift in decision-tree learning from data streams. In: **Proceedings of the Fourth European Symposium on Intelligent Technologies and their implementation on Smart Adaptive Systems**. Aachen, Germany: Verlag Mainz, 2004. p. 218–225.

GÉRON, A. **Mãos à Obra: Aprendizado de Máquina com Scikit-Learn TensorFlow - Conceitos, Ferramentas e Técnicas Para a Construção de Sistemas Inteligentes**. Rio de Janeiro, RJ: Alta Books, 2019. 267-269 p.

GHASEMI, A.; ZAHEDIASL, S. Normality tests for statistical analysis: A guide for non-statisticians. **International Journal of Endocrinology and Metabolism**, v. 10(2), p. 486–489, abr. 2012.

GRUS, J. **Data Science do Zero: primeiras regras com o Python**. 1. ed. Rio de Janeiro: Alta Books, 2016.

GUPTA, L. **Comparison of Hyperparameter Tuning algorithms: Grid search, Random search, Bayesian optimization**. 2020. Disponível em: <https://medium.com/analytics-vidhya/comparison-of-hyperparameter-tuning-algorithms-grid-search-random-search-bayesian-optimization-5326aaef1bd1>. Acesso em: 2 out. 2021.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. H. **The elements of statistical learning: data mining, inference, and prediction**. 2 ed. New York, NY: Springer, 2009.

HAWKINS, D. M. **Identification of Outliers**. Netherlands: Dordrecht: Springer, 1980. 1-3 p.

HEIN, H. **Solar atinge novo marco histórico no Brasil: 13 GW de capacidade instalada**. 2022. Disponível em: <https://canalsolar.com.br/solar-atinge-novo-marco-historico-no-brasil-o-de-13-gw-de-capacidade-instalada/>. Acesso em: 6 jan. 2022.

HEINEN, E. D. **Redes neurais recorrentes e XGBoost aplicados à previsão de radiação solar no horizonte de curto prazo**. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de São Carlos, São Carlos, SP, 2018.

HONDA, H. **Introdução Básica à Clusterização**. 2017. Disponível em: <https://lamfo-unb.github.io/2017/10/05/Introducao_basica_a_clusterizacao/>. Acesso em: 15 ago. 2021.

KALOGIROU, S.; SENCAN, A. Artificial intelligence techniques in solar energy applications. In: MANYALA, R. (Ed.). **Solar Collectors and Panels**. Rijeka: IntechOpen, 2010. cap. 15. Disponível em: <https://doi.org/10.5772/10343>.

KAMAROUTHU, P. **Solar Irradiance Prediction Using Xg-boost With the Numerical Weather Forecast**. Dissertação (Master of Science - MS) – Utah State University, Logan, Utah, 2020.

KOHAVI, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. **International Joint Conferences on Artificial Intelligence**, v. 14, p. 1137–1143, 1995.

KUMARI, P.; TOSHNIWAL, D. Deep learning models for solar irradiance forecasting: A comprehensive review. **Journal of Cleaner Production**, v. 318, p. 128566, 10 out. 2021.

LESOUPLE, J.; BAUDOIN, C.; SPIGAI, M.; TOURNERET, J.-Y. Generalized isolation forest for anomaly detection. **Pattern Recognition Letters**, Elsevier, v. 149, p. 109–119, 2021.

LIN, Y.; KOPRINSKA, I.; RANA, M.; TRONCOSO, A. Pattern sequence neural network for solar power forecasting. **Communications in Computer and Information Science Neural Information Processing**, p. 727–737, 2019.

LIU, F. T.; TING, K. M.; ZHOU, Z.-H. Isolation forest. In: **2008 Eighth IEEE International Conference on Data Mining**. [*S. l.*: *s. n.*], 2008. p. 413–422.

MACHADO, A.; JÚNIOR, L.; NUNES, M. XGBoost na Previsão da Geração de Energia Elétrica em Parques Eólicos. In: **Anais da XIV Conferência Brasileira sobre Qualidade da Energia Elétrica**. Galoa, 2021. ISBN 9786589463061. Disponível em: <https://proceedings.science/proceedings/100186/_papers/130586>.

MAJID, R.; MIR, S. Advances in statistical forecasting methods: An overview. **Economic Affairs**, v. 63, p. 815–831, dez. 2018.

MAKRIDAKIS, S.; SPILIOTIS, E.; ASSIMAKOPOULOS, V. Statistical and machine learning forecasting methods: Concerns and ways forward. **PLoS ONE**, v. 13(3), p. e0194889, mar. 2018.

MALIK, F. **What Is Grid Search?** 2020. Disponível em: <https://medium.com/fintechexplained/what-is-grid-search-c01fe886ef0a>. Acesso em: 31 ago. 2021.

MARTÍNEZ-ÁLVAREZ, F.; TRONCOSO, A.; RIQUELME, J.; AGUILAR-RUIZ, J. Energy time series forecasting based on pattern sequence similarity. **IEEE Transactions on Knowledge and Data Engineering**, v. 23, n. 8, p. 1230–1243, 2011.

MAYER, M. J.; GRÓF, G. Extensive comparison of physical models for photovoltaic power forecasting. **Applied Energy**, Elsevier, v. 283, p. 116239, 02 2021.

MELLIT, A.; PAVAN, A. M. A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected pv plant at trieste, italy. **Solar Energy**, Elsevier, v. 84, p. 807–821, 05 2010.

MENSI, A.; BICEGO, M. Enhanced anomaly scores for isolation forests. **Pattern Recognition**, Elsevier, v. 120, p. 108115, 2021.

MOISEN, G. G. Classification and regression trees. **In: Jørgensen, Sven Erik; Fath, Brian D. (Editor-in-Chief). Encyclopedia of Ecology**, Elsevier, v. 1, p. 582–588, 2008.

NEGNEVITSKY, M. **Artificial intelligence: a guide to intelligent systems**. 2 ed. Harlow, England: Addison-Wesley, 2005. 165-168 p.

NWAIGWE, K. N.; MUTABILWA, P.; DINTWA, E. An overview of solar power (PV systems) integration into electricity grids. **Materials Science for Energy Technologies**, v. 2, p. 629–633, dez. 2019.

OLIVEIRA, R. F. de. **Inteligência artificial**. Londrina, Paraná: Editora e Distribuidora Educacional S.A, 2018. 173-174 p.

OLIVER, M.; JACKSON, T. Energy and economic evaluation of building-integrated photovoltaics. **Energy**, v. 26, p. 431–439, abr. 2001.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V.; VANDERPLAS, J.; PASSOS, A.; COURNAPEAU, D.; BRUCHER, M.; PERROT, M.; DUCHESNAY, E. Scikit-learn: Machine learning in Python. **Journal of Machine Learning Research**, v. 12, p. 2825–2830, 2011.

PEREIRA, E.; MARTINS, F.; COSTA, R.; GONçALVES, A.; LIMA, F.; RÜTHER, R.; ABREU, S.; TIEPOLO, G.; PEREIRA, S.; SOUZA, J. **Atlas Brasileiro de Energia Solar – 2ª Edição**. [*S. l.*: *s. n.*], 2017. ISBN 978-85-17-00089-8.

RAZA, M. Q.; NADARAJAH, M.; EKANAYAKE, C. On recent advances in PV output power forecast. **Solar Energy**, v. 136, p. 125–144, out. 2016.

RIBEIRO, S. **Usinas solares de grande porte atingem 4 GW**. 2021. Disponível em: <https://revistapotencia.com.br/portal-potencia/energia/usinas-solares-de-grande-porte-atingem-4-gw/>. Acesso em: 21 jan. 2022.

SHAPIRO, S. S.; WILK, M. B. An analysis of variance test for normality (complete samples). **Biometrika**, v. 52(3/4), p. 591–611, 1965.

SHENDE, M.; BOKDE, N. **PSF_py: forecasting of univariate time series using the Pattern Sequence-based Forecasting (PSF) algorithm**. GitHub, 2019. Disponível em: <https://github.com/Mayur1009/PSF_py>.

SILVA, I. N.; SPATTI, D. H.; FLAUZINO, R. A. **Redes neurais artificiais para engenharia e ciências aplicadas**. São Paulo: Artliber Editora, 2010.

SIQUEIRA, D. **Histograma: O que é, Exemplos, Gráficos e Tipos**. 2021. Disponível em: <https://www.alura.com.br/artigos/o-que-e-um-histograma>. Acesso em: 16 nov. 2021.

SMITI, A. A critical overview of outlier detection methods. **Computer Science Review**, Elsevier, v. 38, p. 100306, 2020.

SNOEK, J.; LAROCHELLE, H.; ADAMS, R. P. Practical bayesian optimization of machine learning algorithms. **arXiv:1206.2944 [cs, stat]**, 29 ago. 2012. Disponível em: <https://arxiv.org/abs/1206.2944>.

SOBRI, S.; KOOHI-KAMALI, S.; RAHIM, N. A. Solar photovoltaic generation forecasting methods: a review. **Energy Conversion and Management**, Elsevier, v. 156, p. 459–497, 01 2018.

THEODORIDIS, S.; KOUTROUMBAS, K. **Pattern recognition**. Boston: Academic Press, 2003. v. 2. 1 p.

VANDERPLAS, J. **Python Data Science Handbook: essentials tools for working with data**. Sebastopol, CA: O'Reilly Media, 2017. v. 1.

WANG, Z.; KOPRINSKA, I.; RANA, M. Solar power forecasting using pattern sequences. In: **Artificial Neural Networks and Machine Learning – ICANN**. [*S. l.*]: Springer International Publishing, 2017. (Lecture Notes in Computer Science), p. 486–494. ISBN 978-3-319-68611-0.

XU, R.; WUNSCH, D. Survey of clustering algorithms. **IEEE Transactions on Neural Networks**, v. 16(3), p. 645–678, 2005.

YAP, B. W.; SIM, C. H. Comparisons of various types of normality tests). **Journal of Statistical Computation and Simulation**, v. 81(12), p. 2141–2155, 1 dez. 2011.

APÊNDICE A – CÓDIGOS-FONTES DOS ALGORITMOS

Os códigos-fontes utilizados para desenvolvimento do trabalho não foram incluídos no texto em razão da elevada quantidade dos mesmos, que comprometeria o tamanho final do arquivo, bem como a legibilidade dos códigos. Desta forma, disponibilizou-se os arquivos no endereço eletrônico: https://www.dropbox.com/sh/q6aqv7xsqdti0e8/AADVamAj3pi-lxnoFTDVcm0ra?dl=0.