

INSTITUTO FEDERAL DE SANTA CATARINA
CURSO DE ENGENHARIA ELÉTRICA

FRANCISCO EDUARDO GAZANIGA

**ANÁLISE DE DADOS E APRENDIZADO DE MÁQUINA APLICADOS
NA PREVISÃO DE GERAÇÃO DE POTÊNCIA ELÉTRICA DE UMA
PLANTA FOTOVOLTAICA**

TRABALHO DE CONCLUSÃO DE CURSO

ITAJAÍ
2022

FRANCISCO EDUARDO GAZANIGA

**ANÁLISE DE DADOS E APRENDIZADO DE MÁQUINA APLICADOS
NA PREVISÃO DE GERAÇÃO DE POTÊNCIA ELÉTRICA DE UMA
PLANTA FOTOVOLTAICA**

Trabalho de Conclusão de Curso apresentado ao Curso de Engenharia Elétrica do Instituto Federal de Santa Catarina - IFSC Campus Itajaí, como requisito parcial para a obtenção do título de Bacharel em Engenharia Elétrica.

Orientador: Ênio dos Santos Silva
Instituto Federal de Santa Catarina

Coorientador: Jéssica de Aguiar França

ITAJAÍ
2022

AGRADECIMENTOS

Agradeço primeiramente aos meus pais (Francisco de Assis Gazaniga e Sandra Regina Lima) por todo o suporte que me foi dado durante todo o meu período de formação e vida. Sem eles, a possibilidade de me formar em engenharia jamais seria possível. Agradeço a minha parceira, Rafaela Peixer que sempre me incentivou e me acompanhou durante o desenvolvimento do presente trabalho.

Agradeço ao meu orientador Prof. Ênio dos Santos Silva e minha coorientadora Profa. Jéssica de Aguiar França pela disponibilidade de me orientar e pela oportunidade que me foi concedida de poder trabalhar com eles. Agradeço principalmente ao professor Ênio por todo seu auxílio durante o desenvolvimento do trabalho, se demonstrou sempre solícito e me ajudou com todas minhas dúvidas e dificuldades, sem ele a conclusão deste trabalho teria sido muito mais difícil. Agradeço aos meus colegas de curso por todo o apoio durante esta jornada de estudos. Agradeço a todos os professores do curso de Engenharia Elétrica do Instituto Federal de Santa Catarina - Campus Itajaí, os quais tive o prazer de ter tido aula, cada um pôde agregar na minha formação e experiência de vida. Agradeço também, ao Instituto Federal de Santa Catarina - Campus Itajaí.

Cada um de nós é, sob uma perspectiva cósmica, precioso. Se um humano discorda de você, deixe-o viver. Em cem bilhões de galáxias, você não vai achar outro como ele. (SAGAN, Carl).

RESUMO

Atualmente nota-se uma crescente demanda do mercado por soluções que usem aprendizado de máquina (AM), além disso, nos últimos anos o uso da energia fotovoltaica vem se popularizando, principalmente, graças a incentivos dos governos em busca de fontes limpas de energia. Esses dois temas são tópicos ativos de pesquisa na comunidade científica e encontram diversas aplicações na indústria. Nesse contexto, utilizando análise exploratória de dados em conjunto com algoritmos de AM, o presente trabalho desenvolve uma estrutura de códigos (*framework*) destinada à estimação de séries temporais correspondentes à previsão de geração de potência elétrica em sistemas fotovoltaicos (plantas fotovoltaicas). Para a indústria, tal previsão tem sido um desafio comercial, visto a sua dependência quanto a variáveis climáticas, intensidade de irradiação, temperatura, dentre outros parâmetros de difícil previsibilidade, dessa forma, o uso de algoritmos de AM tem se tornado uma importante alternativa para a obtenção de previsões satisfatórias. Particularmente, este trabalho de pesquisa investiga a previsão de geração de potência elétrica realizada por algoritmos de AM baseados em modelos lineares e de redes neurais a partir de dados de plantas fotovoltaicas. Resultados dos desempenhos dos algoritmos de AM são apresentados e avaliados com vistas ao erro quadrático médio da estimação da geração de potência elétrica em corrente alternada (CA), confirmando a eficácia do *framework* desenvolvido neste trabalho.

Palavras-chave: Aprendizado de máquina, energia fotovoltaica, rede neural com perceptron multicamadas, rede neural recorrente com memória longa de curto prazo, regressão linear, regressão polinomial.

ABSTRACT

Currently, there is a growing market demand for solutions that use machine learning (ML), in addition, in recent years the use of photovoltaic energy has become popular, mainly thanks to government incentives in search of clean sources of energy. These two themes are active research topics in the scientific community and find diverse applications in industry. In this context, using exploratory data analysis in conjunction with ML algorithms, the present work develops a code structure (framework) for the estimation of time series corresponding to the forecast of electric power generation in photovoltaic systems (photovoltaic plants). For the industry, such forecast has been a commercial challenge, given its dependence on climatic variables, irradiation intensity, temperature, among other parameters that are difficult to predict, thus, the use of ML algorithms has become an important alternative for obtaining satisfactory forecasts. Particularly, this research work investigates the forecast of electric power generation performed by ML algorithms based on linear models and neural networks from data from photovoltaic plants. Results of the performance of ML algorithms are presented and evaluated with a view to the mean square error of the estimation of electric power generation in alternating current (AC), confirming the effectiveness of the framework developed in this work.

Keywords: Machine learning, photovoltaic energy, neural network multilayer perceptron, neural network long short-term memory, linear regression, polynomial regression.

LISTA DE FIGURAS

Figura 1 – Consumo de energia global por fonte.	2
Figura 2 – Três tipos característicos de curvas de distribuição de frequência.	9
Figura 3 – Os componentes básicos de um neurônio.	13
Figura 4 – Modelo não-linear de um neurônio.	14
Figura 5 – Função sigmoide para parâmetro de inclinação a variável.	15
Figura 6 – Exemplo de RNA multicamadas típica.	16
Figura 7 – Rede neural recorrente.	18
Figura 8 – Funcionamento de uma RNN padrão, contém uma única camada.	19
Figura 9 – Funcionamento de uma LSTM padrão, contém quatro camadas.	20
Figura 10 – Funcionamento de uma LSTM padrão, célula de memória.	20
Figura 11 – Funcionamento de uma LSTM padrão, primeira etapa, esquecer informação.	21
Figura 12 – Funcionamento de uma LSTM padrão, segunda etapa, memória.	21
Figura 13 – Funcionamento de uma LSTM padrão, terceira etapa, atualizar informações.	22
Figura 14 – Funcionamento de uma LSTM padrão, quarta etapa, filtrar informações.	23
Figura 15 – Quantidade de linhas no conjunto de dados para cada inversor fotovoltaico (azul: 3.259 linhas de dados, laranja: 3.195 e vermelho: 2.355).	28
Figura 16 – Potência AC e DC total gerada em um intervalo de 6 horas para todos os 34 dias.	29
Figura 17 – % de potência DC convertida em potência AC.	29
Figura 18 – Irradiação média em um intervalo de 6 horas para todos os 34 dias.	30
Figura 19 – Temperatura média em um intervalo de 6 horas para todos os 34 dias.	30
Figura 20 – Histograma com a simetria dos dados, antes da tratativa dos valores nulos.	32
Figura 21 – Potência AC gerada durante o dia para todos os inversores.	33
Figura 22 – Potência AC gerada durante o dia para todos os inversores (6h — 18h).	34
Figura 23 – Irradiação média para quando a geração de potência foi nula.	35
Figura 24 – Histograma com a simetria dos dados, após tratativa dos valores nulos.	36
Figura 25 – Índices de correlação entre os atributos.	39
Figura 26 – Diagrama de caixa dos atributos.	42
Figura 27 – Potência AC gerada pela planta solar durante os 34 dias.	43
Figura 28 – Perdas (MSE) para MLP com uma única variável de entrada.	47
Figura 29 – Perdas (MSE) para MLP com múltiplas variáveis de entrada	47

Figura 30 – Perdas (MSE) para MLP com método <i>Window</i> e uma única variável de entrada.	48
Figura 31 – Perdas (MSE) para MLP com método <i>Window</i> e múltiplas variáveis de entrada.	48
Figura 32 – Perdas (MSE) para LSTM com uma única variável de entrada. . . .	49
Figura 33 – Perdas (MSE) para LSTM com múltiplas variáveis de entrada. . . .	49
Figura 34 – Perdas (MSE) para LSTM com método <i>Window</i> e uma única variável de entrada.	50
Figura 35 – Perdas (MSE) para LSTM com método <i>Window</i> e múltiplas variáveis de entrada.	51
Figura 36 – Desempenho no teste para a Regressão Linear com uma única variável de entrada.	53
Figura 37 – Previsão para os 34 dias com modelo de Regressão Linear com uma única variável de entrada.	54
Figura 38 – Desempenho no teste para a regressão polinomial com uma única variável de entrada.	55
Figura 39 – Previsão para os 34 dias com modelo de regressão polinomial com uma única variável de entrada.	56
Figura 40 – Desempenho no teste para a Regressão Linear com múltiplas variáveis de entrada.	56
Figura 41 – Previsão para os 34 dias com modelo de Regressão Linear com múltiplas variáveis de entrada.	57
Figura 42 – Desempenho no teste para a regressão polinomial com múltiplas variáveis de entrada.	58
Figura 43 – Previsão para os 34 dias com modelo de regressão polinomial com múltiplas variáveis de entrada.	59
Figura 44 – Desempenho no teste para a MLP com uma única variável de entrada.	60
Figura 45 – Previsão para os 34 dias com modelo de rede neural MLP com uma única variável de entrada.	61
Figura 46 – Desempenho no teste para a MLP com método <i>Window</i> e uma única variável de entrada.	61
Figura 47 – Previsão para os 34 dias com modelo de rede neural MLP com método <i>Window</i> e uma única variável de entrada.	62
Figura 48 – Desempenho no teste para a MLP com múltiplas variáveis de entrada.	63
Figura 49 – Previsão para os 34 dias com modelo de rede neural MLP com múltiplas variáveis de entrada.	64
Figura 50 – Desempenho no teste para a MLP com método <i>Window</i> e múltiplas variáveis de entrada.	64

Figura 51 – Previsão para os 34 dias com modelo de rede neural MLP com método <i>Window</i> e múltiplas variáveis de entrada.	65
Figura 52 – Desempenho no teste para a LSTM com uma única variável de entrada.	66
Figura 53 – Previsão para os 34 dias com modelo de rede neural LSTM com uma única variável de entrada.	67
Figura 54 – Desempenho no teste para a LSTM com método <i>Window</i> e uma única variável de entrada.	67
Figura 55 – Previsão para os 34 dias com modelo de rede neural LSTM com método <i>Window</i> e uma única variável de entrada.	68
Figura 56 – Desempenho no teste para a LSTM com múltiplas variáveis de entrada.	69
Figura 57 – Previsão para os 34 dias com modelo de rede neural LSTM com múltiplas variáveis de entrada.	70
Figura 58 – Desempenho no teste para a LSTM com método <i>Window</i> e múltiplas variáveis de entrada.	70
Figura 59 – Previsão para os 34 dias com modelo de rede neural LSTM com método <i>Window</i> e múltiplas variáveis de entrada.	71
Figura 60 – Comparativo entre o atributo de entrada e de saída do modelo com uma única variável de entrada.	73
Figura 61 – Potência AC gerada nos dias 15/05/20 a 17/05/20 sem tratamento. .	81
Figura 62 – Potência AC gerada nos dias 18/05/20 a 21/05/20 sem tratamento. .	82
Figura 63 – Potência AC gerada nos dias 22/05/20 a 25/05/20 sem tratamento. .	83
Figura 64 – Potência AC gerada nos dias 26/05/20 a 29/05/20 sem tratamento. .	84
Figura 65 – Potência AC gerada nos dias 30/05/20 a 02/06/20 sem tratamento. .	85
Figura 66 – Potência AC gerada nos dias 03/06/20 a 06/06/20 sem tratamento. .	86
Figura 67 – Potência AC gerada nos dias 07/06/20 a 10/06/20 sem tratamento. .	87
Figura 68 – Potência AC gerada nos dias 11/06/20 a 14/06/20 sem tratamento. .	88
Figura 69 – Potência AC gerada nos dias 15/06/20 a 17/06/20 sem tratamento. .	89
Figura 70 – Potência AC gerada nos dias 15/05/20 a 17/05/20 com tratamento. .	90
Figura 71 – Potência AC gerada nos dias 18/05/20 a 21/05/20 com tratamento. .	91
Figura 72 – Potência AC gerada nos dias 22/05/20 a 25/05/20 com tratamento. .	92
Figura 73 – Potência AC gerada nos dias 26/05/20 a 29/05/20 com tratamento. .	93
Figura 74 – Potência AC gerada nos dias 30/05/20 a 02/06/20 com tratamento. .	94
Figura 75 – Potência AC gerada nos dias 03/06/20 a 06/06/20 com tratamento. .	95
Figura 76 – Potência AC gerada nos dias 07/06/20 a 10/06/20 com tratamento. .	96
Figura 77 – Potência AC gerada nos dias 11/06/20 a 14/06/20 com tratamento. .	97
Figura 78 – Potência AC gerada nos dias 15/06/20 a 17/06/20 com tratamento. .	98

LISTA DE TABELAS

Tabela 1 – Atributos da geração de energia solar.	25
Tabela 2 – Atributos do clima.	26
Tabela 3 – Coeficiente de de assimetria de Pearson (<i>skewness</i>), antes de tratativas de valores nulos.	31
Tabela 4 – Coeficiente de de assimetria de Pearson (<i>skewness</i>), após tratativas de valores nulos.	37
Tabela 5 – Correlação em relação ao atributo alvo.	38
Tabela 6 – Fator de inflação de variância para os atributos independentes. . . .	40
Tabela 7 – Melhores fatores de inflação de variância para os atributos independentes.	40
Tabela 8 – Relatório OLS.	41
Tabela 9 – Hiperparâmetros utilizados no modelo MLP padrão.	46
Tabela 10 – Hiperparâmetros utilizados no modelo MLP com método <i>Window</i> . . .	47
Tabela 11 – Hiperparâmetros utilizados no modelo LSTM padrão.	49
Tabela 12 – Hiperparâmetros utilizados no modelo LSTM com método <i>Window</i> . .	50
Tabela 13 – MSE e RMSE obtidos com a Regressão Linear para a previsão dos dados de treino, teste e amostra completa.	53
Tabela 14 – MSE e RMSE obtidos com a Regressão Polinomial para a previsão dos dados de treino, teste e amostra completa.	55
Tabela 15 – MSE e RMSE obtidos com a Regressão Linear com múltiplas variáveis para a previsão dos dados de treino, teste e amostra completa.	57
Tabela 16 – Resultados obtidos com a Regressão Polinomial com múltiplas variáveis para a previsão dos dados de treino, teste e amostra completa.	58
Tabela 17 – MSE e RMSE obtidos com a rede neural MLP com uma única variável para a previsão dos dados de treino, teste e amostra completa.	60
Tabela 18 – MSE e RMSE obtidos com a rede neural MLP com método <i>Window</i> e uma única variável para a previsão dos dados de treino, teste e amostra completa.	62
Tabela 19 – MSE e RMSE obtidos com a rede neural MLP para múltiplas variáveis para a previsão dos dados de treino, teste e amostra completa.	63
Tabela 20 – MSE e RMSE obtidos com a rede neural MLP com método <i>Window</i> para múltiplas variáveis para a previsão dos dados de treino, teste e amostra completa.	65
Tabela 21 – MSE e RMSE com a rede neural LSTM com uma única variável para a previsão dos dados de treino, teste e amostra completa.	66

Tabela 22 – RMSE e RMSE obtidos com a rede neural LSTM com método <i>Window</i> e uma única variável para a previsão dos dados de treino, teste e amostra completa.	68
Tabela 23 – MSE e RMSE obtidos com a rede neural LSTM para múltiplas variáveis para a previsão dos dados de treino, teste e amostra completa.	69
Tabela 24 – MSE e RMSE obtidos com a rede neural LSTM com método <i>Window</i> para múltiplas variáveis para a previsão dos dados de treino, teste e amostra completa.	71
Tabela 25 – Erro quadrático médio normalizado obtido para todos os modelos treinados com múltiplas variáveis de entrada.	72
Tabela 26 – Erro quadrático médio normalizado obtido para todos os modelos treinados com uma única variável de entrada.	73
Tabela 27 – Raíz do erro quadrático médio obtido para todos os modelos treinados para a amostra de teste.	74

LISTA DE ABREVIATURAS E SIGLAS

AC	<i>Alternating Current</i>
AED	<i>Análise Exploratória de Dados</i>
AM	Aprendizado de máquina
ANFIS	<i>Adaptive Network based Fuzzy Inference System</i>
BPNN	<i>Back Propagation Neural Network</i>
BRT	<i>Bagged Regression Trees</i>
CA	Corrente Alternada
CC	Corrente Contínua
DC	<i>Direct Current</i>
EDA	<i>Exploratory Data Analysis</i>
ELM	<i>Extreme Learning Machine</i>
ENN	<i>Elman Neural Network</i>
GABPNN	<i>BPNN with Genetic Algorithm Optimization</i>
GRNN	<i>Generalized Regression Neural Network</i>
IA	Inteligência Artificial
KNN	<i>K Nearest Neighbours</i>
LSTM	<i>Long Short-Term Memory</i>
MLP	<i>Multilayer Perceptron</i>
MLR	<i>Multiple Linear Regression</i>
MSE	<i>Mean Squared Error</i>
NARXNN	<i>Non-linear Auto Regressive Neural Network with Exogenous Inputs</i>
NN	Neural Networks
REQM	Raíz do Erro Quadrático Médio
RF	<i>Random Forest</i>
RMSE	<i>Root Mean Squared Error</i>
RNA	Redes Neurais Artificiais
SVR	<i>Support Vector Regression</i>
TIM	<i>Total Irradiance Monitor</i>
VIF	<i>Variance inflation factor</i>

LISTA DE SÍMBOLOS

As	Coeficiente de assimetria de Pearson
\bar{x}	Média
Mo	Moda
σ	Desvio padrão
r	Correlação
VIF	Fator de inflação da variância
$Q1$	Primeiro quartil
$Q2$	Mediana
$Q3$	Terceiro quartil
IQ	Distância interquartil
$\hat{y}^{(i)}$	Saída prevista
$x_j^{(i)}$	j-ésimo atributo de entrada da i-ésima amostra
b	Viés
v_k	Soma das saídas geradas pelas sinapses
$\varphi(v)$	Função sigmoide
a	Inclinação da função sigmoide
t	Tempo
\tanh	Tangente hiperbólica
h_t	Saída de uma rede neural
\bar{C}_t	Cópia de uma célula de memória de uma rede neural
C_t	Célula de memória de uma rede neural
o_t	Portão de saída
R^2	Coeficiente de determinação
QE_{reta}	Diferença entre os pontos de uma reta
y	Saída
$RMSE$	Raíz do erro quadrático médio

SUMÁRIO

1	INTRODUÇÃO	1
1.1	ORGANIZAÇÃO DO TRABALHO	3
2	OBJETIVOS	4
2.1	OBJETIVO GERAL	4
2.2	OBJETIVOS ESPECÍFICOS	4
3	JUSTIFICATIVA	5
4	REFERENCIAL TEÓRICO	6
4.1	ENERGIA SOLAR	6
4.2	PREVISÃO DE GERAÇÃO DA ENERGIA SOLAR	6
4.3	ANÁLISE EXPLORATÓRIA DE DADOS	8
4.3.1	DISTRIBUIÇÃO DE FREQUÊNCIA	9
4.3.2	CORRELAÇÃO	9
4.3.2.1	MULTICOLINEARIDADE	9
4.3.3	QUARTIL	10
4.4	REGRESSÃO LINEAR	10
4.5	REGRESSÃO POLINOMIAL	11
4.6	REDES NEURAIS	12
4.6.1	NEURÔNIO BIOLÓGICO	12
4.6.2	NEURÔNIO ARTIFICIAL	14
4.6.3	REDES NEURAIS ARTIFICIAIS - RNA	15
4.6.3.1	PERCEPTRON MULTICAMADAS - MLP	17
4.6.3.2	ALGORITMO DE RETROPROPAGAÇÃO	17
4.6.4	REDES NEURAIS RECORRENTES - RNN	18
4.6.4.1	MEMÓRIA LONGA DE CURTO PRAZO - LSTM	19
4.7	MÉTRICAS PARA AVALIAR UM MODELO	23
4.7.1	COEFICIENTE DE DETERMINAÇÃO - R^2	23
4.7.2	RAÍZ DO ERRO QUADRÁTICO MÉDIO - RMSE	24
5	METODOLOGIA	25
5.1	ANÁLISE EXPLORATÓRIA	26
5.1.1	PREPARAÇÃO DOS DADOS	27
5.1.2	VISUALIZAÇÃO DOS DADOS	28
5.1.3	ANÁLISE DE ASSIMETRIA	30
5.1.4	DEFINIÇÃO DO ATRIBUTO ALVO	37

5.1.5	ANÁLISE DE CORRELAÇÃO	38
5.1.6	ANÁLISE DE MULTICOLINEARIDADE	39
5.1.7	ANÁLISE DE REGRESSÃO LINEAR	41
5.1.8	ANÁLISE DE QUARTIL	42
5.2	MODELAGEM	43
5.2.1	DIVISÃO DA AMOSTRA	43
5.2.2	MODELOS LINEARES	44
5.2.2.1	MODELO REGRESSÃO LINEAR	44
5.2.2.2	MODELO REGRESSÃO LINEAR COM MULTIVARIÁVEIS	44
5.2.2.3	MODELO REGRESSÃO POLINOMIAL COM MULTIVARIÁVEIS	45
5.2.3	MODELOS DE REDE NEURAL	45
5.2.3.1	MÉTODO VANILLA	45
5.2.3.2	MÉTODO WINDOW	45
5.2.3.3	MODELO MLP	46
5.2.3.4	MODELO MLP COM MÉTODO WINDOW	47
5.2.3.5	MODELO LSTM	48
5.2.3.6	MODELO LSTM COM MÉTODO WINDOW	50
6	ANÁLISE E DISCUSSÃO DOS RESULTADOS	52
6.1	PREVISÃO COM MODELOS LINEARES	52
6.1.1	PREVISÃO REGRESSÃO LINEAR	52
6.1.2	PREVISÃO REGRESSÃO POLINOMIAL	54
6.1.3	PREVISÃO REGRESSÃO LINEAR COM MULTIVARIÁVEIS	56
6.1.4	PREVISÃO REGRESSÃO POLINOMIAL COM MULTIVARIÁVEIS	57
6.2	PREVISÃO COM REDES NEURAS	59
6.2.1	PREVISÃO MLP	59
6.2.2	PREVISÃO MLP MÉTODO WINDOW	61
6.2.3	PREVISÃO MLP COM MULTIVARIÁVEIS	62
6.2.4	PREVISÃO MLP MÉTODO WINDOW COM MULTIVARIÁVEIS	64
6.2.5	PREVISÃO LSTM	65
6.2.6	PREVISÃO LSTM MÉTODO WINDOW	67
6.2.7	PREVISÃO LSTM COM MULTIVARIÁVEIS	68
6.2.8	PREVISÃO LSTM MÉTODO WINDOW COM MULTIVARIÁVEIS	70
6.3	COMPARAÇÃO DOS MODELOS APLICADOS	71
7	CONCLUSÃO E CONSIDERAÇÕES FINAIS	75
7.1	TRABALHOS FUTUROS	75
7.2	CONSIDERAÇÕES FINAIS	76

Referências	77
Apêndices	80
APÊNDICE A Potência AC gerada para cada um dos 34 dias, sem tratamento para nulos	81
APÊNDICE B Potência AC gerada para cada um dos 34 dias, com tratamento para nulos	90

1 INTRODUÇÃO

A inteligência artificial (IA) vem se popularizando desde 1970 na solução de problemas reais, inicialmente através de sistemas especialistas, que baseados em regras lógicas, tentavam reproduzir a tomada de decisão de profissionais de determinadas áreas de conhecimento. Com o passar do tempo e com o aprimoramento de hardwares destinados a processamento digital de sinais, a solução de problemas cada vez mais complexos utilizando IA tornou-se usual (FACELI et al., 2021).

A melhora na coleta, armazenamento e processamento de dados, e ainda, o aumento de trabalhadores especializados (embora que ainda em baixo número), permitiu que empresas adotassem a inteligência artificial para aplicação em diversos setores do seu negócio.

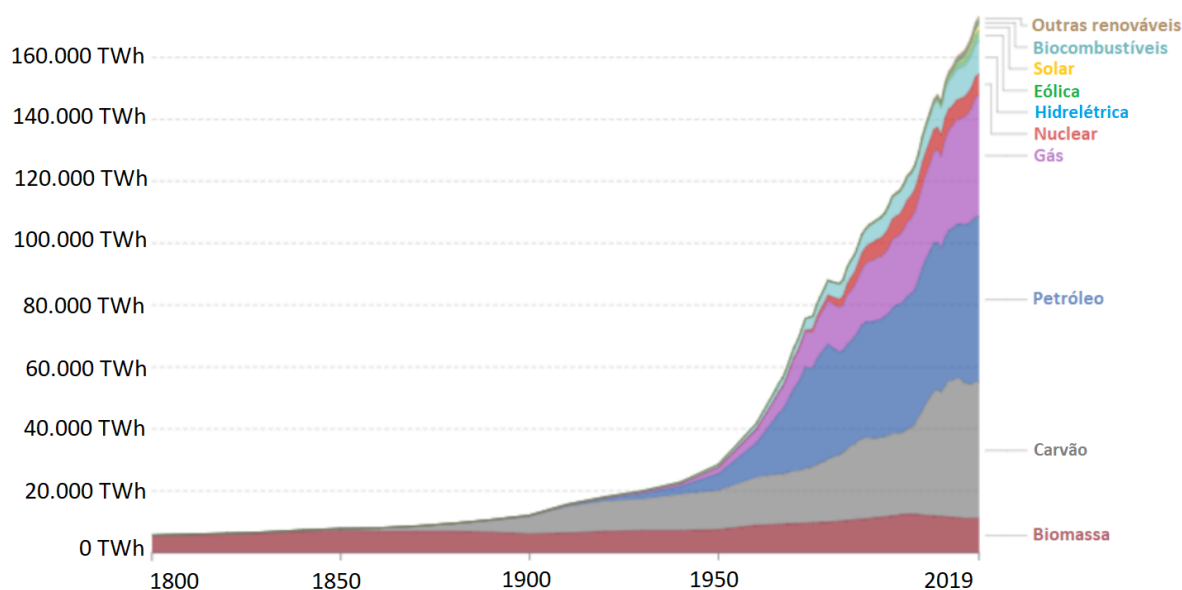
Uma subárea da IA que está sendo adotada em diversas aplicações na indústria é o aprendizado de máquina (AM), utilizado tanto em estratégias de negócio, como também em aplicações que oferecem melhores funcionalidades para o usuário, legendas automáticas para vídeos no Youtube, por exemplo. As diversas aplicações do AM podem ser divididas em duas categorias principais: aprendizado supervisionado e não supervisionado, sendo o primeiro para atividades preditivas e o segundo para atividades descritivas (FACELI et al., 2021).

As atividades preditivas consistem em atividades onde se deseja, com base em atributos de entrada, prever um atributo alvo, por exemplo, prever o valor da fatura de energia para os próximos meses, com base no histórico de consumo de energia de uma residência. Nesse contexto, o AM pode ser aplicado na previsão da geração de energia de um sistema fotovoltaico.

Nos últimos anos o uso da energia fotovoltaica vem se popularizando, principalmente graças aos incentivos do governo e a demanda crescente por fontes limpas de energia. Além disso, os painéis fotovoltaicos evoluíram tecnologicamente, apresentando um maior rendimento e confiabilidade, sem falar na flexibilidade proporcionada pelo sistema, permitindo que seja instalado de diversas formas e em diferentes espaços (VIAN et al., 2021).

Essa popularização pode ser observada com o gráfico da Figura 1, que apresenta o consumo de energia em todo o mundo do ano de 1800 até 2019. Em 2010 a energia solar apresentava o menor consumo dentre todas as fontes de energia, com apenas 88 terawatt-hora, quase uma década depois ela já apresenta relevância maior que os biocombustíveis e outras energias renováveis, com um consumo de 1.793 terawatt-hora.

Figura 1 – Consumo de energia global por fonte.



Fonte: Adaptado de (RITCHIE; ROSER, 2020)

A previsão da geração de potência em um sistema fotovoltaico auxilia não só na manutenção do sistema, como também em um melhor aproveitamento do recurso energético, o que permite prever quando e em que período se exigirá mais do sistema, auxiliando ainda, na criação de políticas de racionamento de energia. Com uma previsão adequada, pode-se estimar, por exemplo, quando será necessário realizar investimentos na rede de distribuição (VIAN et al., 2021).

Realizar a previsão da geração de potência elétrica em um sistema fotovoltaico é uma tarefa complicada, visto a sua dependência quanto a variáveis climáticas, estação do ano, intensidade de irradiação, e temperatura, parâmetros de difícil previsibilidade.

Portanto, para uma previsão satisfatória, o uso de algoritmos de AM em conjunto com uma grande amostra de dados torna-se uma solução interessante para problemas de estimação de séries temporais em sistemas de energia (FRACCANABIA; MARIANI, 2021). Particularmente, o estado da arte da área de AM apresenta diversos modelos para a realização de previsões (estimação de séries temporais) e tomadas de decisão com base no histórico de um dado sinal ou sistema. Especificamente, este trabalho aplica modelos (utilizando AM) destinados à estimação de séries temporais, em sistemas fotovoltaicos para a previsão de geração de energia, com base na potência elétrica em corrente alternada (CA) resultante.

Dessa forma, neste trabalho de conclusão de curso, usando um conjunto de dados estruturados de duas plantas fotovoltaicas localizadas na Índia, a terceira maior geradora de energia do mundo (RITCHIE; ROSER, 2020), técnicas de regressão (usando modelos lineares e redes neurais) serão aplicadas a sistemas fotovoltaicos, visando responder a seguinte pergunta: os modelos de AM estudados apresentam

um desempenho satisfatório na previsão da geração de potência elétrica em corrente alternada de uma planta fotovoltaica?

1.1 ORGANIZAÇÃO DO TRABALHO

Inicialmente, no Capítulo 4 será apresentado um breve panorama da energia solar e o uso de algoritmos de AM aplicados para previsões de séries temporais. Ainda neste capítulo serão tratados de forma introdutória os principais temas que norteiam o desenvolvimento do trabalho, sendo eles a análise exploratória (Seção 4.3), regressão linear (Seção 4.4), redes neurais (Seção 4.6) e métricas objetivas de avaliação dos modelos usados nos algoritmos de AM (Seção 4.7).

A etapa de desenvolvimento do *framework* de AM é tratada no Capítulo 5, onde será explicado como foram definidos os atributos de entrada e saída, como foi feita a divisão da amostra e como foram organizados os treinamentos dos modelos de AM. Por fim, no Capítulo 6 são apresentados e discutidos os resultados obtidos com as previsões realizadas, efetivando a conclusão no Capítulo 7.

2 OBJETIVOS

2.1 OBJETIVO GERAL

Investigar modelos de aprendizado de máquina para realizar a previsão da geração de potência elétrica de uma planta fotovoltaica.

2.2 OBJETIVOS ESPECÍFICOS

- Estudar modelos de aprendizado de máquina;
- Realizar a análise exploratória dos dados;
- Desenvolver uma estrutura de códigos (*framework*) para a investigação e estimação de séries temporais referentes a geração de potência elétrica de uma planta fotovoltaica;
- Realizar a previsão da geração de potência elétrica de uma planta fotovoltaica;
- Avaliar resultados obtidos de modo a verificar se a previsão foi satisfatória.

3 JUSTIFICATIVA

Mesmo com a pandemia do COVID-19, a energia solar apresentou um crescimento estável entre os anos de 2019 e 2020 (IEA, 2020). Atualmente o Brasil vem enfrentando uma crise energética devido à escassez de chuvas, o que fez com que as hidrelétricas, a maior fonte de energia do país, tivessem seus reservatórios a níveis muito baixos, podendo vir a propiciar ainda mais o crescimento das energias renováveis.

A crise energética em pauta e a preocupação global com o meio ambiente, tornam as energias renováveis destaques em um contexto energético, e neste contexto a energia solar é protagonista. Por vir se destacando e ganhando relevância, é importante que esta tecnologia se torne cada vez mais eficiente e viável, e uma forma de viabilizar e validar a eficiência da energia solar, é realizando previsões de qualidade.

Realizar a previsão de potência elétrica de maneira satisfatória de um sistema fotovoltaico garante segurança e confiabilidade para a rede elétrica (BASTOS et al., 2020). Utilizar de modelos de AM para a finalidade supracitada, incentiva o uso de tecnologias de sistemas fotovoltaicos a pouco tempo no mercado e que vêm se desenvolvendo exponencialmente.

Além disso, até o momento o Curso de Engenharia Elétrica do Instituto Federal de Santa Catarina, Câmpus Itajaí, não apresenta uma disciplina de AM, tema circundante deste trabalho. Logo, sua apresentação pode ampliar o horizonte dos alunos para esta área que possui inúmeras aplicações na engenharia elétrica.

4 REFERENCIAL TEÓRICO

4.1 ENERGIA SOLAR

A geração de energia elétrica através de radiação solar é possível graças a tecnologia fotovoltaica, onde placas solares compostas por células fotovoltaicas, convertem a energia proveniente da luz do sol em eletricidade, através do efeito fotovoltaico.

Por depender da radiação proveniente do sol, a energia solar é uma fonte de energia muito dependente de fatores climáticos, e por este motivo enfrenta limitações peculiares, como em períodos do dia onde a geração é interrompida devido à falta de incidência solar, além de situações de sombreamento, onde nuvens impedem a passagem dos raios solares.

Apesar de tais limitações, a facilidade na instalação do sistema, a valorização do imóvel onde o sistema é instalado, o rápido retorno do alto investimento inicial e o baixo impacto ambiental, acabam por compensar o uso desta tecnologia, além de que a intermitência da energia pode ser contornada com o uso de baterias para o armazenamento da energia gerada.

Devido ao clima tropical e posição geográfica, o Brasil pode ser considerado um país ideal para a implementação da tecnologia fotovoltaica. Apesar disso, de acordo com um estudo realizado pela *International Energy Agency Photovoltaic Power Systems Programme* com dados do ano de 2020, o Brasil não está nem entre os dez países com maior índice de energia solar instalada, no entanto, no ano de 2020 figurou a nona posição entre os países que mais instalaram sistemas fotovoltaicos (IEA, 2021).

4.2 PREVISÃO DE GERAÇÃO DA ENERGIA SOLAR

Para realizar previsões e análises com base em dados, pode-se utilizar de técnicas de inteligência artificial com base em algoritmos de aprendizado de máquina. Tais algoritmos consistem em um conjunto de fórmulas matemáticas utilizadas para obter uma determinada saída.

Treinar um modelo de aprendizado de máquina para realizar previsões não é uma tarefa simples e costuma ser um processo de treinamento exaustivo em busca da minimização da taxa de erro sobre um determinado conjunto de dados (SU; BATZELIS; PAL, 2019). Na literatura, diversos métodos de previsão já foram utilizados para prever a geração de energia fotovoltaica, dentre eles se tem métodos estatísticos, de inteligência artificial, métodos físicos e híbridos, que combinam um ou mais métodos (ABDEL-NASSER; MAHMOUD, 2019).

Prever a geração de potência elétrica de uma planta fotovoltaica é uma tarefa

complicada, devido às incertezas na previsão da irradiação solar, que podem variar de acordo com problemas de sombreamento, chuvas, temperatura da bateria, entre outros fatores que reduzem a precisão das previsões. Para contornar esse problema, o uso de modelos envolvendo Redes Neurais Artificiais (RNA) torna-se interessante, pois apresenta um desempenho satisfatório na previsão de séries temporais. Para uma maior precisão é essencial uma quantidade abundante de dados, quanto maior a quantidade disponível e melhor a qualidade destes dados, melhor será a previsão realizada. O pré tratamento realizado e a padronização dos dados são também relevantes para um aumento na precisão do resultado (LIU et al., 2017).

Liu et al. (2017) utilizou um modelo de rede neural conhecido como *Back Propagation Neural Network* (BPNN), para realizar a previsão de potência elétrica de um sistema fotovoltaico localizado em Ashland, nos EUA, resultando em uma previsão com ótima precisão e eficiência do sistema solar, com 24 horas de antecedência. Este e outros modelos de rede neural são comparados no artigo de Su, Batzelis e Pal (2019), na tentativa de prever a geração de potência de uma planta de 406 MW localizada em Norfolk na Inglaterra. Além da já citada BPNN, o artigo compara o desempenho de outras cinco redes neurais, sendo a *Non-linear Auto Regressive Neural Network with Exogenous Inputs* (NARXNN), *Elman Neural Network* (ENN), *Generalized Regression Neural Network* (GRNN), *Adaptive Network based Fuzzy Inference System* (ANFIS) e *BPNN with Genetic Algorithm Optimization* (GABPNN).

Além das seis redes neurais, o artigo ainda tenta prever a geração de potência elétrica com outros quatro algoritmos de aprendizado de máquina, são eles os modelos de regressão *K Nearest Neighbours* (KNN), *Extreme Learning Machine* (ELM), *Random Forest* (RF) e *Support Vector Regression* (SVR). Por fim, é criado ainda um modelo híbrido, realizando a combinação dos três modelos que obtiveram melhor desempenho. Ao comparar todos os modelos testados, isoladamente, a rede neural NARXNN obteve o melhor desempenho geral, no entanto, os algoritmos de regressão foram mais robustos, não havendo tanta disparidade entre si, dentre os algoritmos de regressão, o melhor desempenho foi do RF. A melhor previsão, no entanto, foi realizada pelo algoritmo híbrido, o que mostra que utilizar uma combinação de várias técnicas é mais eficiente do que utilizar um único modelo (SU; BATZELIS; PAL, 2019).

Um modelo popular para se realizar previsões no curto prazo é a rede neural recorrente com memória longa de curto prazo [*Long Short-Term Memory* (LSTM)], Torres et al. (2021) constata que num período de curto prazo a LSTM apresenta uma melhor precisão. Na geração fotovoltaica, Abdel-Nasser e Mahmoud (2019) constataram que a LSTM, quando comparada com os modelos *Multiple Linear Regression* (MLR), *Bagged Regression Trees* (BRT) e rede neural convencional [*Neural Networks* (NN)], apresenta um menor erro de precisão. Além disso esse modelo foi utilizado também na previsão de outros parâmetros em um sistema solar, Bastos et al. (2020) utilizou a

LSTM para prever os dados de corrente e tensão produzidos pelos painéis solares. A LSTM consegue considerar as mudanças temporais, o que é de grande importância para um modelo prever a geração de potência elétrica de um sistema fotovoltaico (ABDEL-NASSER; MAHMOUD, 2019). Dada as considerações supracitadas, faz-se interessante o uso da LSTM no presente trabalho, apenas a título de comparação, de modo a investigar um diferente modelo dos já supracitados, será feito o uso da rede neural perceptron multicamadas [*Multilayer Perceptron* (MLP)], além do uso de modelos lineares, como a regressão linear e polinomial.

4.3 ANÁLISE EXPLORATÓRIA DE DADOS

Com o desenvolvimento dos algoritmos de AM, o aprimoramento dos *hardwares* dedicados ao processamento digital de sinais e a capacidade de armazenamento de dados, a necessidade de conhecimento especializado sobre os dados visando a solução de um determinado problema, foi se tornando cada vez mais substituível por sistemas puramente automáticos. Tais sistemas são tipicamente conhecidos na literatura como sistemas de caixa preta (*black box*), no qual o conhecimento especializado (sobre os dados) de um profissional pode ser dispensado (desde que se disponha de uma grande quantidade de dados e um grande poder computacional). No entanto, para áreas como engenharia, a não utilização do conhecimento específico de uma determinada área não é interessante. Ao invés de uma simples substituição como ocorre nos sistemas *black box*, uma solução alternativa poderia ser a utilização conjunta dos algoritmos de AM com uma análise crítica dos dados por parte de um especialista da área em questão. Nesse contexto, tal análise crítica é realizada e discutida na etapa de análise exploratória.

Para que haja uma melhor compreensão e análise crítica dos dados disponíveis em um conjunto de treinamento, validação e teste, a análise exploratória de dados (AED) [*Exploratory Data Analysis* (EDA)] é uma etapa fundamental e uma das que mais consome tempo de execução, é nela que são identificadas as variáveis de entrada e saída, é realizado o tratamento de valores faltantes e extremos (que desviam muito do padrão) com o intuito de verificar se existe alguma anomalia, medir a dispersão dos dados e verificar a dependência entre os atributos com cálculo de correlação. Além disso, podem ser formuladas hipóteses a serem validadas, para se obter um melhor entendimento dos dados, e por fim criar ou realizar a transformação de variáveis caso necessário.

A AED é uma área relativamente nova na estatística, surgiu em 1977 no livro "*Exploratory Data Analysis*" de John W. Tukey, autor responsável por criar laços com comunidades de engenharia e ciência da computação (BRUCE; BRUCE, 2019). A seguir serão apresentados alguns métodos estatísticos que serão utilizados na AED deste trabalho de conclusão de curso.

4.3.1 DISTRIBUIÇÃO DE FREQUÊNCIA

Verificar a distribuição da frequência em um conjunto de dados antes de aplicá-lo em um modelo de AM é extremamente útil para entender a distribuição dos dados e detectar padrões de cada atributo da amostra. Quando a distribuição é simétrica, esta pode apresentar um formato de sino (gaussiano), como o da Figura 2, no entanto, nem toda distribuição simétrica é gaussiana.

Figura 2 – Três tipos característicos de curvas de distribuição de frequência.



Fonte: (GUPTA; GUTTMAN, 2017)

Nesse contexto, o coeficiente de assimetria de Pearson (A_s) pode ser utilizado como indicativo para representar a distribuição da frequência dos dados. Tal coeficiente pode ser expresso pela Equação (1), onde \bar{x} representa a média da distribuição dos dados, Mo representa a moda¹ e σ representa o desvio padrão dos dados em relação a média.

$$A_s = \frac{\bar{x} - Mo}{\sigma} \quad (1)$$

4.3.2 CORRELAÇÃO

A correlação (r) mede o grau de relacionamento entre duas variáveis lineares (x e y), variando de -1 a 1, onde quanto mais próximo de 1 ou -1, mais correlacionada são as variáveis, ou seja, maior é a relação linear entre elas. O coeficiente de correlação pode ser calculado pela Equação (2).

$$r = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (2)$$

4.3.2.1 MULTICOLINEARIDADE

A multicolinearidade visa investigar a dependência entre os valores de entrada do modelo, caso a dependência seja forte, se diz que existe multicolinearidade, por

¹Valor com maior frequência absoluta em um conjunto de dados.

consequente se tem que estas variáveis são redundantes para estimar o valor a ser previsto.

Uma das maneiras de se detectar a multicolinearidade entre duas variáveis é através do fator de inflação da variância [*Variance inflation factor* (VIF)]. Um fator considerado aceitável por alguns autores gira em torno de valores menores do que 4 ou 5, enquanto outros afirmam que valores que excederem 10 irão causar efeito sobre os coeficientes de regressão (MILOCA; CONEJO, 2008). O VIF pode ser calculado pela expressão da Equação (3).

$$VIF = \frac{1}{1 - r^2} \quad (3)$$

4.3.3 QUARTIL

Utilizando a teoria de quartil em conjunto com diagramas de caixa (*box plot*), torna-se fácil de identificar valores *outliers* em uma amostra de dados, valores que divergem consideravelmente da média, além de propiciar um maior entendimento da distribuição dos dados.

Em um quartil os dados são divididos em percentis, P_{25} , P_{50} e P_{75} , onde são, respectivamente, o primeiro quartil ($Q1$), a mediana ($Q2$) e o terceiro quartil ($Q3$). O primeiro quartil indica os 25% valores menores, com a mediana os 50% valores centrais e o terceiro quartil apresenta os 25% maiores valores (SICSÚ; DANA, 2012).

A distância entre o primeiro e terceiro quartil é denominada distância interquartil (IQ), sendo uma medida de dispersão dos dados e apresenta uma ideia da dispersão dos valores centrais da distribuição.

4.4 REGRESSÃO LINEAR

O modelo de regressão linear utiliza uma hipótese linear (equação da reta) para a estimação de séries em geral. Para essa estimação, o modelo de regressão linear realiza uma soma ponderada dos valores de entrada (combinação linear), conforme Equação (4), onde $\hat{y}^{(i)}$ é o valor previsto da i -ésima amostra, n é a quantidade de atributos de entrada (n determina a dimensão do espaço de entrada), $x_j^{(i)}$ é o j -ésimo atributo de entrada da i -ésima amostra, w_j é o j -ésimo coeficiente angular (peso) do modelo e b representa o coeficiente linear (viés) do modelo (GÉRON, 2021).

$$\hat{y}^{(i)} = b + w_1x_1^{(i)} + w_2x_2^{(i)} + \dots + w_nx_n^{(i)} \quad (4)$$

A Equação (5) mostra a notação vetorial da regressão linear.

$$\hat{y}^{(i)} = w_0 + [w_1 \dots w_n] \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}^{(i)} = b + \mathbf{w}^T \mathbf{x}^{(i)} \quad (5)$$

, onde $b = w_0$ e $\mathbf{w} = [w_1 \dots w_n]^T$.

Matematicamente, é conveniente considerar $b = 0$ e incluir um valor de entrada constante $x_0 = 1$, dessa forma pode-se representar os atributos de entrada como $\mathbf{x}^{(i)} = [1 \ x_1 \ \dots \ x_n]^{(i)}$ e os pesos do modelo como $\mathbf{w}^T = [w_0 \ w_1 \ \dots \ w_n]^T$. Assim, a Equação (5) é reescrita como:

$$\hat{y}^{(i)} = \mathbf{w}^T \mathbf{x}^{(i)} \quad (6)$$

Considerando uma função custo $J(\mathbf{w})$, pode-se calcular o valor ótimo de w no sentido de minimização do erro quadrático médio (MSE) (entre o valor previsto \hat{y} com o valor verdadeiro y). Assim:

$$J(\mathbf{w}) = \frac{1}{m} \sum_{i=1}^m (\hat{y}^{(i)} - y^{(i)})^2 = \frac{1}{m} \sum_{i=1}^m (\mathbf{w}^T \mathbf{x}^{(i)} - y^{(i)})^2 \quad (7)$$

Onde m representa a quantidade total de amostras no conjunto de treinamento.

Então, para calcular o valor de w que minimize a função custo $J(\mathbf{w})$, faz-se $\frac{\partial J(\mathbf{w})}{\partial w_j} = 0$. Tal procedimento é conhecido como método dos mínimos quadrados (*ordinary least squares*) e tem como solução a equação normal reescrita aqui na Equação (8). Esta é uma função de característica analítica e que pode apresentar um ótimo desempenho para um baixo volume de dados, visto que a inversão de matrizes da equação exige um grande poder computacional.

$$\mathbf{w}_{\text{ótimo}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{y} \quad (8)$$

Para a regressão linear usando uma única variável (unidimensional) tem-se como saída uma simples reta, já para a regressão linear usando múltiplas variáveis (multidimensional) obtêm-se um hiperplano no espaço n dimensional de dados (GUPTA; GUTTMAN, 2017).

4.5 REGRESSÃO POLINOMIAL

Poucas observações do cotidiano possuem relação linear, para descrever situações não-lineares. A regressão linear usando atributos transformados polinomialmente pode ser uma boa escolha. Basicamente, pode-se aproximar qualquer curva usando funções base (como um polinômio, por exemplo), desde que se encontre a função base adequada que caracterize satisfatoriamente a curva dos dados de interesse. Escolher bons atributos ou transformações de atributos é um problema de engenharia de atributos (*feature engineering*) e é fortemente dependente da AED (discutida na Seção 4.3)

A regressão polinomial é um caso especial da regressão linear com múltiplas variáveis, aqui cada variável de entrada é transformada por um polinômio de grau p

conforme mostrado pela Equação (9).

$$\varphi(\mathbf{x}, p) = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & \dots & x_1^p \\ 1 & x_2 & x_2^2 & x_2^3 & \dots & x_2^p \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & x_n^3 & \dots & x_n^p \end{bmatrix} = \begin{bmatrix} \varphi_1(x,p) \\ \varphi_2(x,p) \\ \vdots \\ \varphi_n(x,p) \end{bmatrix} \quad (9)$$

Em geral, se pode considerar como atributos de entrada $x_j = \varphi_j(x,p)$. Assim, a regressão polinomial (regressão linear com transformação polinomial grau p de atributos) é descrita conforme Equação (10).

$$\hat{y} = \mathbf{w}_1\varphi_1(x,p) + \mathbf{w}_2\varphi_2(x,p) + \dots + \mathbf{w}_n\varphi_n(x,p) \quad (10)$$

Mesmo que contenham termos não-lineares, o modelo de regressão polinomial ainda é considerado um modelo linear, pois estes são lineares em w , considerando as suas correspondentes funções base $\varphi_j(x,p)$ (NAVIDI, 2012).

4.6 REDES NEURAIS

As redes neurais têm o intuito de tentar aproximar (imitar) as conexões e transmissões de informações realizadas pelo cérebro humano para realizar determinadas tarefas. Dessa forma, as conexões realizadas pelo cérebro (conexões neuronais) também são realizadas de forma artificial pelos modelos de redes neurais, armazenando informações e aprendendo com elas.

Uma rede neural é um processador maciçamente paralelamente distribuído constituído de unidades de processamento simples, que têm a propensão natural para armazenar conhecimento experimental e torná-lo disponível para o uso. Ela se assemelha ao cérebro em dois aspectos:

1. O conhecimento é adquirido pela rede a partir de seu ambiente através de um processo de aprendizagem.
2. Forças de conexão entre neurônios, conhecidas como pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido (HAYKIN, 2007, p. 28).

A rede neural possui uma grande capacidade de generalização, ou seja, resultados satisfatórios podem ser obtidos mesmo que a rede opere com dados que não foram “vistos” na etapa de treinamento (dados de entrada diferentes daqueles com os quais a rede neural foi treinada) (HAYKIN, 2007).

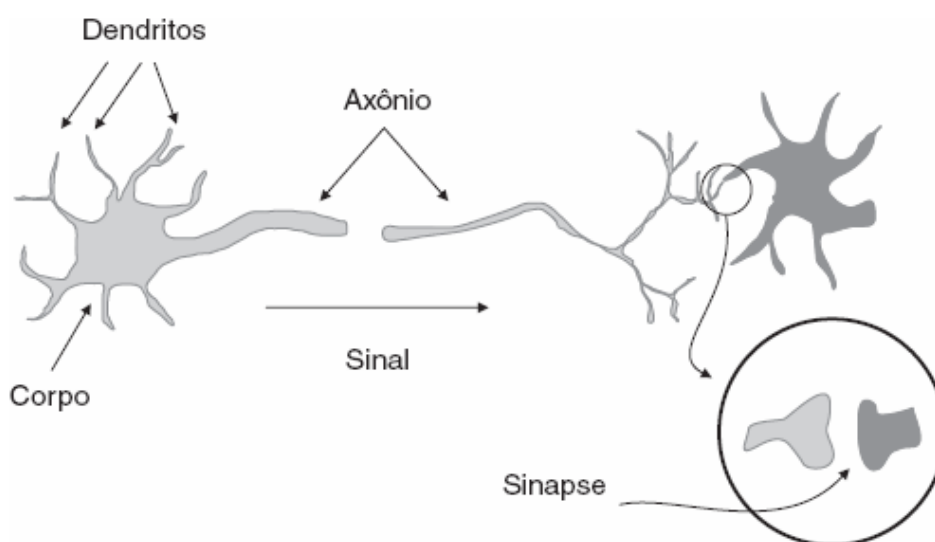
4.6.1 NEURÔNIO BIOLÓGICO

Antes de adentrar no funcionamento do neurônio artificial, é interessante compreender como que funciona um neurônio biológico. Os neurônios tem a função

de realizar o processamento das informações no sistema nervoso. Eventos neurais ocorrem em uma velocidade de milissegundos, o que torna os neurônios biológicos cerca de cinco a seis ordens de grandeza mais lentos do que portas lógicas em silício, que se comunicam na ordem de nanossegundos (HAYKIN, 2007). O que compensa a velocidade menor dos neurônios biológicos, é a sua quantidade enorme com conexões entre si. Estima-se que existam 85 bilhões de neurônios no cérebro humano (BEAR; CONNORS; PARADISO, 2002).

O neurônio (ilustrado na Figura 3) é constituído por uma parte central chamada soma, corpo celular ou pericário, e desta saem diversas conexões, tubos finos conhecidos como neuritos, divididos em dendritos e axônios. O axônio é mais espesso e longo que os dendritos, com um diâmetro uniforme e ramificações em ângulo reto, seria como uma linha de transmissão do neurônio, conduzindo os impulsos nervosos. Para os dendritos, bem menores que os axônios, pode-se fazer uma analogia como sendo as antenas do neurônio, recebendo estímulos (BEAR; CONNORS; PARADISO, 2002).

Figura 3 – Os componentes básicos de um neurônio.



Fonte: (FACELI et al., 2021)

A comunicação entre os neurônios ocorre através das sinapses, estas podem ser elétricas ou químicas, e possuem a função de transmitir informações para os neurônios pelos neuritos. As informações das sinapses são recebidas pelos dendritos (antenas), processadas no corpo celular e transmitidas para outros neurônios percorrendo o axônio (linha de transmissão) até as suas terminações, onde ali encontram as sinapses que intermeiam a conexão de um axônio de um neurônio com os dendritos de outro (ESTATIDADOS, 2019). A maioria das sinapses do cérebro são químicas, podendo ser bidirecional (a mesma sinapse que emite informação também recebe) ou não, as sinapses elétricas, são todas bidirecionais. (BEAR; CONNORS; PARADISO,

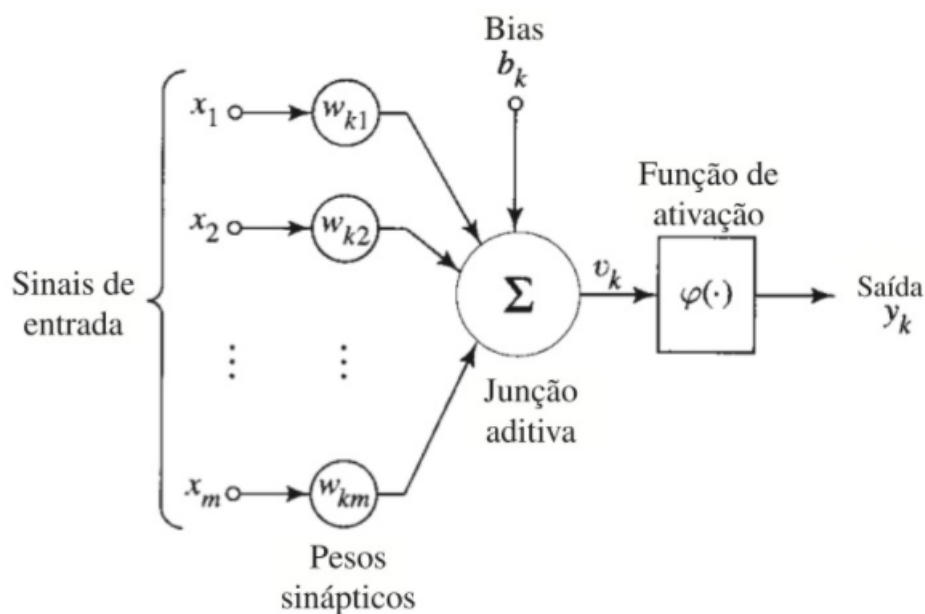
2002).

4.6.2 NEURÔNIO ARTIFICIAL

Um neurônio artificial não consegue replicar a complexidade de um neurônio real, no entanto, representa o que há de mais próximo hoje, em emular o processo de aprendizado em termos matemáticos.

A Figura 4 exemplifica um neurônio artificial, sinais de entrada (x_j) são multiplicados por pesos sinápticos (w_{kj}), o neurônio soma as saídas geradas pelas sinapses (v_k) e aplica em uma função de ativação ($\varphi(\cdot)$), que tem o intuito de restringir a amplitude do sinal de saída do neurônio. Certas funções de ativação permitem que redes neurais modelem padrões não lineares complexos, que simples modelos podem deixar passar (LABS, 2014).

Figura 4 – Modelo não-linear de um neurônio.



Fonte: (HAYKIN, 2007)

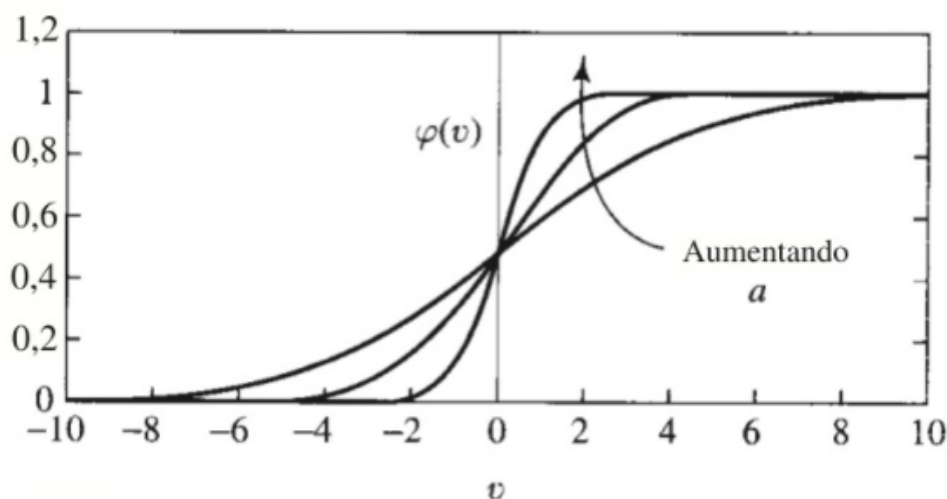
No modelo representando na Figura 4, existe um bias, ou viés (b_k), cuja função é aumentar ou diminuir a entrada líquida da função de ativação (HAYKIN, 2007). Um neurônio pode ser representado pela Equação (11), ao aplicar a função de ativação obtém-se a Equação (12).

$$u_k = \sum_{j=1}^m w_{kj} x_j \quad (11)$$

$$y_k = \varphi(v_k) = \varphi(u_k + b_k) \quad (12)$$

Dentre as funções de ativação mais utilizadas em uma rede neural artificial, pode-se destacar a função sigmoide ($\varphi(v)$) ilustrada na Figura 5. Representada pela Equação (13), $\varphi(v)$ é uma função crescente que exibe um equilíbrio entre comportamento linear e não-linear (HAYKIN, 2007).

Figura 5 – Função sigmoide para parâmetro de inclinação a variável.



Fonte: (HAYKIN, 2007)

$$\varphi(v) = \sigma(v) = \frac{1}{1 + e^{-av}} \quad (13)$$

Onde o parâmetro a representa o valor de inclinação da função sigmoide. Importante notar que a função sigmoide é uma função diferenciável, esta característica é muito importante para a teoria de redes neurais, pois permite o ajuste dos pesos dos neurônios (w) usando derivações parciais.

4.6.3 REDES NEURAS ARTIFICIAIS - RNA

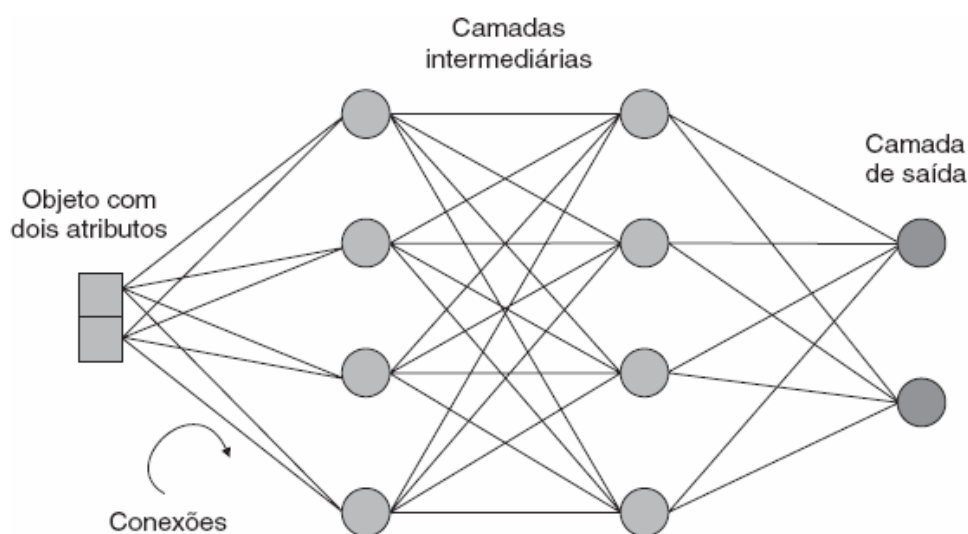
As redes neurais artificiais (RNA) tentam emular o cérebro humano, não são algoritmos que se resumem em realizar determinadas tarefas, como, por exemplo, jogar xadrez, realizar cálculos matemáticos ou automatizar uma tarefa manual. Sim, o cérebro humano consegue realizar tais atividades, mas em um jogo de videogame, por exemplo, ao jogar pela primeira vez o cérebro identifica o que deve ser feito na tentativa e erro, reconhecendo padrões, aprende que o personagem deve pular em determinados momentos, se abaixar em outros, adquire conhecimento, para então resultar em uma saída que é jogar da forma necessária para se completar a partida. Assim é a rede neural artificial, baseada em entradas (conhecimento) ela busca realizar o aprendizado, moldado pelos pesos, se desenvolvendo de forma autônoma, aprendendo a jogar um jogo por si própria ou a realizar previsões de acordo com padrões e cálculos

matemáticos, a classificar objetos conforme aprende a reconhecer padrões em imagens, entre outras aplicações. Basicamente, as redes neurais artificiais são utilizadas para problemas de regressão e classificação.

Uma rede neural artificial é composta por vários neurônios, que estão interconectados por conexões sinápticas (pesos), os pesos podem assumir valores positivos ou negativos sendo ajustados no decorrer do processo de aprendizado, para codificar o conhecimento adquirido pela rede, influenciando na precisão do resultado obtido pela rede neural (FACELI et al., 2021).

Estes neurônios podem estar divididos em uma ou mais camadas, em uma RNA multicamadas típica (Figura 6), por exemplo, tem-se os atributos de entrada conectados aos neurônios da primeira camada e a saída destes conectadas aos neurônios da segunda camada, e isto pode se repetir n vezes, podendo haver quantas camadas intermediárias forem necessárias, estas camadas intermediárias são também denominadas como camadas ocultas (*hidden layers*), por fim se tem a camada de saída que pode ser composta por um ou mais neurônios, que irão resultar em uma ou mais saídas. (BARRETO, 2002).

Figura 6 – Exemplo de RNA multicamadas típica.



Fonte: (FACELI et al., 2021)

Existem diversos tipos de RNA que irão se diferir basicamente em relação a sua arquitetura e aprendizado, a arquitetura sendo o tipo e a quantidade de neurônios utilizados, além da quantidade de camadas e também a forma que estes são conectados entre si, enquanto que o aprendizado se refere às regras que são utilizadas para ajustar os pesos da rede e ao tipo de informação que é utilizada por essas regras (FACELI et al., 2021).

A principal vantagem de uma RNA é a possibilidade de realização de transformações não lineares (via função de ativação, por exemplo: função sigmoid $\varphi(\cdot)$) e assim

“gerar” atributos mais discriminativos para um determinado problema. Analogamente à regressão polinomial, a RNA também usa transformações dos atributos, no entanto realiza tais transformações através da combinação das saídas não lineares de cada camada da rede.

4.6.3.1 PERCEPTRON MULTICAMADAS - MLP

A rede neural perceptron multicamadas, conhecida também como *Multilayer Perceptron* (MLP), é uma RNA que pode ser utilizada para problemas de regressão. É uma RNA composta por uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída.

Na saída de cada neurônio se tem uma função de ativação realizando uma transformação não-linear. Se aqui em cada saída houvesse uma inversão de matriz para minimizar o erro, como ocorre no método dos mínimos quadrados, seria inviável realizar o treinamento, pois exigiria muito processamento computacional, além disso, quando a não-linearidade é aplicada na saída do neurônio a superfície do erro deixa de ser quadrática, logo, a solução ótima determinística representada pela equação normal torna-se difícil de ser encontrada. Por essas razões, na rede neural o modelo irá minimizar o erro utilizando o método do gradiente descendente. Tal método é iterativo e corresponde a uma aproximação da solução ótima, na rede neural a cada iteração o modelo vai ajustando seus pesos em busca do mínimo global na superfície de erro.

Ao ser utilizada para problemas preditivos, de regressão, caso seja utilizada para prever um valor único, a camada de saída passa a ser composta por somente um único neurônio, que será o valor predito (GÉRON, 2021).

4.6.3.2 ALGORITMO DE RETROPROPAGAÇÃO

O algoritmo de treinamento de retropropagação, ou algoritmo de *backpropagation*, foi apresentado em 1986 por David Rumelhart, Geoffrey Hinton e Ronald Williams. Graças a este algoritmo que foi possível treinar as redes MLPs com sucesso. O algoritmo consiste em calcular o valor dos coeficientes w_j de um modelo através do método iterativo do gradiente descendente mostrado na Equação (14),

$$\mathbf{w}^{[t]} = \mathbf{w}^{[t-1]} - \alpha \frac{2}{m} \mathbf{X}^T (\mathbf{X} \mathbf{w}^{[t-1]} - \mathbf{y}) \quad (14)$$

Onde $\mathbf{w}^{[t]}$ representa os coeficientes \mathbf{w} no instante atual t , $\mathbf{w}^{[t-1]}$ representa os coeficientes \mathbf{w} no instante prévio $t - 1$ e α representa a taxa de aprendizagem do modelo (GÉRON, 2021).

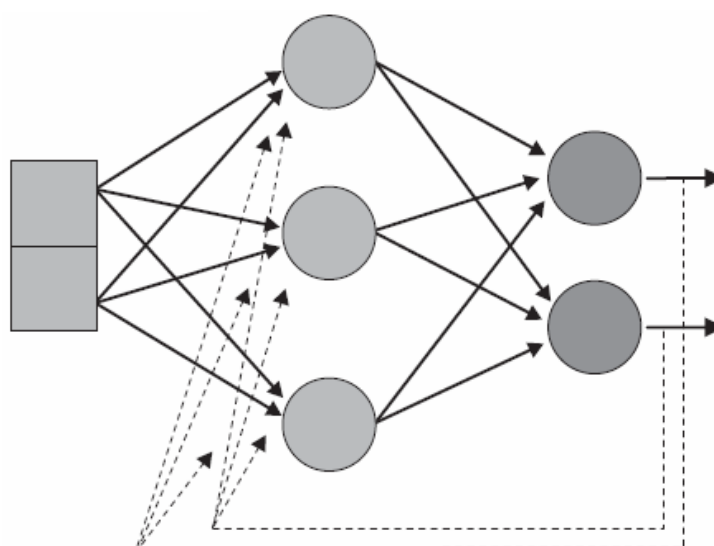
De forma resumida, o algoritmo de retropropagação ajusta o peso das conexões entre os neurônios de cada camada da rede. Para fazer isto, primeiramente é realizada uma previsão (*forward pass*) e calculado o erro, em seguinte é realizado um *reverse*

pass, onde se passa por cada camada no sentido inverso e através de derivadas parciais, calcula-se a contribuição do erro de cada conexão por camada, por fim, os pesos são ajustados no sentido da minimização do erro, para mais detalhes, veja Géron (2021).

4.6.4 REDES NEURAIAS RECORRENTES - RNN

Tipicamente, em uma rede neural, os dados de treinamento fluem de uma forma unidirecional, da camada de entrada até a camada de saída. Nas redes neurais recorrentes (*Recurrent Neural Network* - RNN) ocorre uma retroalimentação, ou seja, a saída de um neurônio pode “alimentar” a entrada de um neurônio da camada anterior (o retroalimentando), conforme mostrado pela Figura 7. Esta característica permite que as RNN consigam lidar com problemas sequenciais e atributos não-lineares.

Figura 7 – Rede neural recorrente.



Fonte: (FACELI et al., 2021)

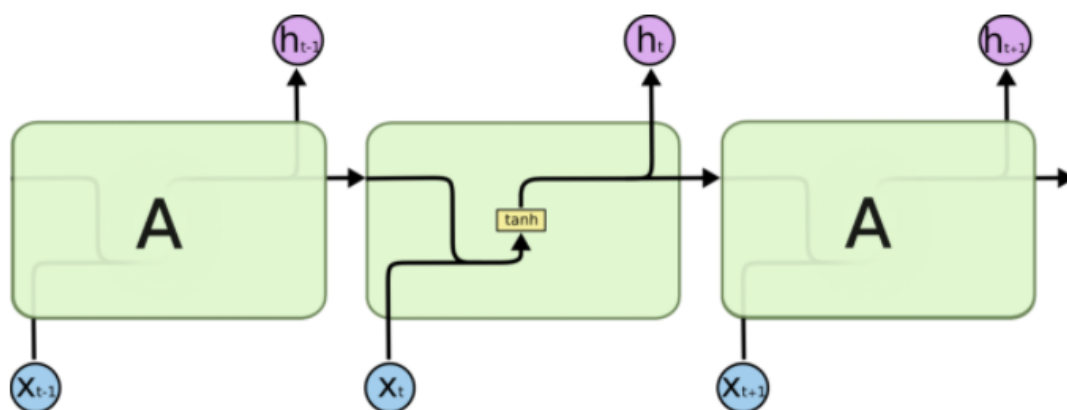
Por conseguirem lidar com problemas sequenciais e atributos não-lineares, as redes neurais recorrentes são utilizadas em problemas de série temporal e na simulação de sistemas dinâmicos (FACELI et al., 2021). Na RNN os neurônios recebem não só as informações de entrada, mas também informações previstas anteriormente, ou seja, previsões já realizadas influenciam nas previsões atuais (BROWNLEE, 2017). Esta é uma característica importante no cérebro humano. Humanos não realizam pensamentos do zero a todo momento, cada pensamento é moldado de acordo com experiências passadas (OLAH, 2015).

Uma RNN pode ser imaginada como uma série em cadeia de várias cópias de uma mesma rede neural, onde cada uma transmite a informação gerada para a

seguinte, ou seja, a rede segue considerando previsões anteriores para as previsões futuras, a sequência temporal das informações é muito importante aqui.

A Figura 8 representa o funcionamento de uma RNN padrão. Basicamente se tem a rede neural no presente momento (t), recebendo as informações de entrada (x_t), que passam por uma única camada de neurônios sob a função de ativação tangente hiperbólica (\tanh), resultando na saída presente (h_t) variando de -1 a 1 (graças a tangente hiperbólica), no entanto, além de receber as informações de entrada, a camada de neurônios presente também recebe as previsões realizadas pela rede no passado (h_{t-1}) e ainda transmite o que previu para a rede futura ($t + 1$), que será recebido com os dados futuros (x_{t+1}), resultando na saída futura (h_{t+1}). Assim, em uma camada no instante t os dados de entrada (x_t) são concatenados com as informações previstas no instante $t - 1$ (h_{t-1}), resultando em um vetor de entrada $[h_{t-1}, x_t]$.

Figura 8 – Funcionamento de uma RNN padrão, contém uma única camada.



Fonte: (OLAH, 2015)

No entanto, a RNN tem dificuldade em lidar com problemas com dependências de longo prazo, ou seja, quando a informação necessária está muito distante do ponto onde ela é requisitada (OLAH, 2015). Para resolver este problema, pode-se utilizar a rede neural *Long Short-Term Memory* (LSTM).

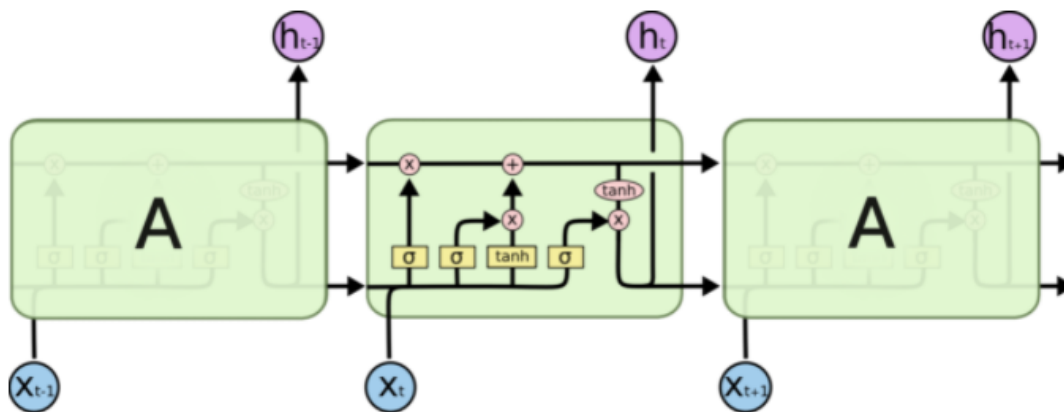
4.6.4.1 MEMÓRIA LONGA DE CURTO PRAZO - LSTM

Introduzida por Hochreiter e Schmidhuber (1997) e posteriormente utilizada e aprimorada por diversos grupos de pesquisa, a rede neural memória longa de curto prazo, mais conhecida como *Long Short-Term Memory* (LSTM) é um tipo de RNN criada para resolver o problema de dependência de longo prazo (OLAH, 2015). Uma característica interessante da LSTM é que ela consegue esquecer e memorizar informações.

A Figura 9 representa o funcionamento de uma LSTM padrão, diferente da RNN padrão que possui apenas uma única camada de neurônio (\tanh) para cada

instante de tempo, aqui se tem quatro camadas de neurônios (blocos destacados em amarelo na Figura 9), três usando funções de ativação sigmoide φ ou σ e uma usando função de ativação \tanh interagindo em cada instante (por exemplo, no instante t).

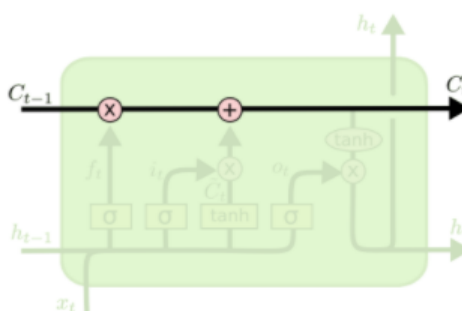
Figura 9 – Funcionamento de uma LSTM padrão, contém quatro camadas.



Fonte: (OLAH, 2015)

A rede LSTM possui uma trilha que corresponde ao gerenciamento do fluxo de informação que é processado em cada instante t . Essa trilha é destacada na Figura 10 e representa uma esteira que percorre toda a rede com algumas interações lineares. A informação poderá ser atualizada através das interações de adição e multiplicação, ou seja, dados poderão ser adicionados na memória e também ponderados ou completamente esquecidos.

Figura 10 – Funcionamento de uma LSTM padrão, célula de memória.

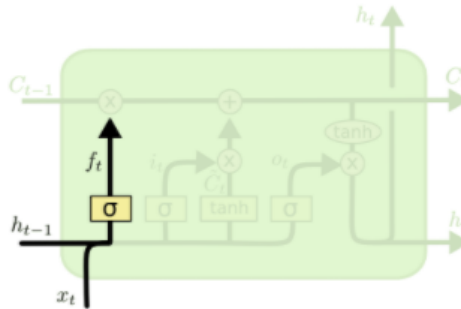


Fonte: (OLAH, 2015)

A primeira camada da LSTM (Figura 11) é composta por uma função sigmoide que conforme visto na Figura 5 varia de 0 a 1, ela irá representar aqui o portão do esquecimento da rede, que terá a função de decidir quanto da informação do vetor $[h_{t-1}, x_t]$ será ignorado, esquecido. A camada recebe como entrada a previsão anterior (h_{t-1}) combinada com as novas informações (x_t), sua saída (f_t) será um número entre 0 e 1, onde 0 significa que a informação será totalmente esquecida e 1 significa que a

informação será completamente mantida. A Equação (15) representa matematicamente o que foi aqui descrito, onde w_f é o peso aplicado sobre o vetor que contém a entrada e previsões anteriores.

Figura 11 – Funcionamento de uma LSTM padrão, primeira etapa, esquecer informação.

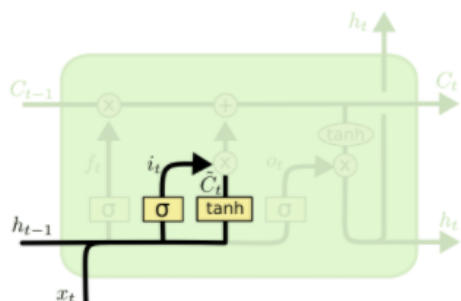


Fonte: (OLAH, 2015)

$$f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f) \tag{15}$$

A segunda etapa (Figura 12) corresponde a memória da rede, tem a funcionalidade de lembrar o que ocorreu em passos anteriores. Para realizar tal feito, na primeira camada o portão de entrada irá receber o vetor $[h_{t-1}, x_t]$ e assim como na primeira etapa, algumas previsões serão lembradas ou esquecidas com o auxílio da camada com função sigmoide, a saída da função sigmoide é denominada como portão de entrada (i_t). Além disso, uma cópia do vetor $[h_{t-1}, x_t]$ é feita e passada para o intervalo de -1 a 1 pela função tangente hiperbólica, resultando em \bar{C}_t . A saída do portão de entrada (i_t) é ponderada com (\bar{C}_t) através da multiplicação de ambos (ROHRER, 2017).

Figura 12 – Funcionamento de uma LSTM padrão, segunda etapa, memória.



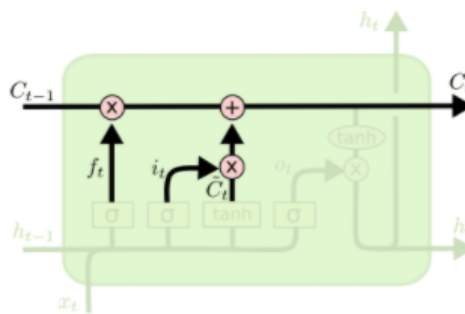
Fonte: (OLAH, 2015)

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i) \tag{16}$$

$$\bar{C}_t = \tanh(w_C \cdot [h_{t-1}, x_t] + b_C) \quad (17)$$

Escolhidos os valores a serem esquecidos e lembrados, nesta terceira etapa (Figura 13) será efetivamente atualizada a informação que consta na célula de memória (C_{t-1}) para se tornar uma nova célula (C_t). O antigo estado é multiplicado por f_t , esquecendo o que foi selecionado na primeira etapa, e somado às informações selecionadas para serem lembradas $i_t \cdot \bar{C}_t$ (Equação (18)).

Figura 13 – Funcionamento de uma LSTM padrão, terceira etapa, atualizar informações.

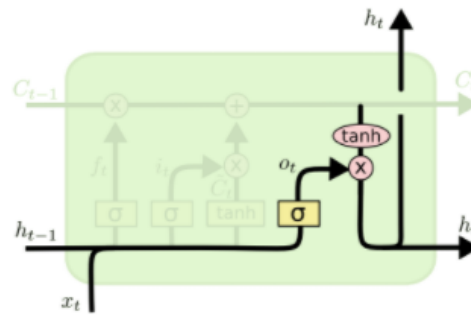


Fonte: (OLAH, 2015)

$$C_t = C_{t-1} \cdot f_t + \bar{C}_t \cdot i_t \quad (18)$$

Por fim, é decidido quais previsões se tornarão a nova saída (h_t), a camada com função sigmoide irá selecionar, baseado em previsões passadas e em novas informações, quais previsões devem seguir adiante e quais devem se manter na memória da rede, as previsões passadas (o_t) são multiplicadas pelas informações a serem lembradas que constam na célula (C_t), antes passando por uma função tangente hiperbólica, de modo a transformar os valores em um intervalo entre -1 e 1, evitando que a saída possua valores exorbitantes, visto que as informações somadas na célula na etapa anterior, podem resultar em valores muito maiores do que 1 e/ou muito menores do que -1 (OLAH, 2015).

Figura 14 – Funcionamento de uma LSTM padrão, quarta etapa, filtrar informações.



Fonte: (OLAH, 2015)

$$o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o) \quad (19)$$

$$h_t = o_t \cdot \tanh C_t \quad (20)$$

Resumindo o funcionamento da LSTM, temos que ela é composta por três portões principais, o portão do esquecimento, de entrada e de saída. Basicamente, as informações previstas anteriormente e os novos dados percorrerão o portão do esquecimento, que irá fazer com que algumas das informações armazenadas na memória da rede sejam esquecidas através da multiplicação ($f_t \cdot C_{t-1}$). O portão de entrada irá lembrar de informações consideradas relevantes, terá tais informações reguladas e então acrescentadas a memória da rede pela adição (Equação (18)). Por fim, os itens da memória da rede serão ponderados e multiplicados pelo vetor com entradas e previsões anteriores, resultando na nova previsão.

4.7 MÉTRICAS PARA AVALIAR UM MODELO

Existem algumas métricas para avaliar se um modelo de regressão foi bem sucedido na previsão dos dados, dentre elas se tem o coeficiente de determinação (R^2), o erro quadrático médio (MSE) e a raiz do erro quadrático médio (RMSE).

4.7.1 COEFICIENTE DE DETERMINAÇÃO - R^2

O coeficiente de determinação irá calcular o quanto o modelo criado é melhor em relação a um modelo que representa simplesmente a média dos valores. Sua resposta irá variar de 0 a 1, a título de exemplo, um coeficiente de determinação igual a 0,87, significa que o modelo representa 87% da variância dos dados. Um R^2 igual a 1, indica que o modelo conseguiu prever 100% da variância dos dados, logo, conseguiu passar por todos os dados realizando a previsão correta (TECH, 2019).

Para calcular o coeficiente de determinação, basta realizar o cálculo da Equação (21), onde QE_{reta} é a diferença (erro) entre os pontos da reta prevista para a reta que representa os dados reais, e QE_{media} é a diferença (erro) entre os pontos da média dos dados reais até os pontos que representam a previsão.

$$R^2 = 1 - \frac{QE_{reta}}{QE_{media}} \quad (21)$$

4.7.2 RAÍZ DO ERRO QUADRÁTICO MÉDIO - RMSE

A Raíz do Erro Quadrático Médio (REQM), conhecido também como *Root Mean Squared Error* (RMSE), é comumente utilizado para medir a acurácia de modelos aplicados na meteorologia, qualidade do ar, e pesquisas de cunho climático (CHAI; DRAXLER, 2014).

O uso do RMSE é mais apropriado em casos onde a distribuição do erro tende a apresentar uma curva Gaussiana (CHAI; DRAXLER, 2014). Ele calcula o somatório da raiz quadrática média dos erros entre os valores observados (y) e previstos (\hat{y}), conforme Equação (22).

$$RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2} \quad (22)$$

Quanto menor o valor do RMSE, mais satisfatório é o modelo previsto. Na prática, avaliar um modelo com *Mean Squared Error* (MSE) ou RMSE leva ao mesmo resultado, pois o valor que minimiza sua função também minimiza sua raiz quadrada (GÉRON, 2021).

5 METODOLOGIA

Os dados do presente trabalho foram obtidos de um conjunto de dados fornecido por Kannal (2020), no site *Kaggle*, uma comunidade *online* de cientistas de dados e profissionais de aprendizado de máquina.

Os atributos que serão analisados, obtidos dos inversores fotovoltaicos, estão reunidos na Tabela 1. Para atingir o objetivo de previsão da geração de potência elétrica do sistema fotovoltaico, se tentará prever o atributo de potência elétrica em corrente alternada de nome “AC_POWER”.

Tabela 1 – Atributos da geração de energia solar.

Atributo	Descrição	Unidade
DATE_TIME	Data no formato dia-mês-ano hora:minuto:segundo.	-
PLANT_ID	Número de identificação da planta solar.	-
SOURCE_KEY	Número de identificação do inversor.	-
DC_POWER	Potência elétrica em corrente contínua gerada pelo inversor em um intervalo de 15 minutos.	kW
AC_POWER	Potência elétrica em corrente alternada gerada pelo inversor em um intervalo de 15 minutos.	kW
DAILY_YIELD	Potência acumulada pelo inversor até o momento naquele dia.	-
TOTAL_YIELD	Rendimento total do inversor até aquele momento.	-

Fonte: Autor, 2021

Os dados referentes ao clima, foram obtidos por dois sensores, cada um dedicado apenas para cada planta solar. O sensor foi acoplado em um único painel solar da planta, os atributos dos dados estão listados na Tabela 2.

Tabela 2 – Atributos do clima.

Atributo	Descrição	Unidade
DATE_TIME	Data no formato dia-mês-ano hora:minuto:segundo.	-
PLANT_ID	Número de identificação da planta solar.	-
SOURCE_KEY	Número de identificação do sensor acoplado em um painel solar.	-
AMBIENT_TEMPERATURE	Temperatura ambiente na planta solar.	°C
MODULE_TEMPERATURE	Temperatura no painel solar ao qual o sensor está acoplado.	°C
IRRADIATION	Quantidade de irradiação para o intervalo de 15 minutos.	W/m ²

Fonte: Autor, 2021

Ao estudar os dados da planta 1, percebe-se que a potência elétrica em corrente alternada possui um valor 10 vezes maior do que a potência elétrica em corrente contínua (CC), o que é claramente um erro, visto que a eficiência de conversão de potência em CC para CA em um sistema fotovoltaico é de aproximadamente 90%, além disso, os dados de geração da planta 1 possuem uma divergência no atributo "DATE_TIME" em comparação aos outros grupos de dados, neste o atributo de data apresenta um padrão de hora:minuto, ao invés de hora:minuto:segundo. Por estes motivos, determinou-se que a análise de dados será realizada somente sobre a planta 2.

Na Terra, o valor de irradiação solar total é de aproximadamente $1.360 \pm 0,5 \text{ W/m}^2$, de acordo com aferições realizadas em 2008 pelo *Total Irradiance Monitor* (TIM) da NASA (KOPP; LEAN, 2011), o que indica que os valores de irradiação coletados para o atributo "IRRADIATION" provavelmente estão relacionados com a área da superfície do sensor, que deve ser próxima da unidade de cm^2 , o que resultaria em uma irradiação 10.000 vezes menor do que a irradiação presente na localidade em que as plantas solares estão localizadas, o que explica sua magnitude de baixo valor.

Os atributos de rendimento não necessariamente se referem ao rendimento dos inversores, o que é o caso do atributo "DAILY_YIELD", que representa um acumulativo da potência gerada no dia.

5.1 ANÁLISE EXPLORATÓRIA

É muito difícil obter alguma informação útil de um conjunto de dados sem realizar previamente uma análise exploratória dos mesmos, por isso, antes de solucionar o problema proposto serão realizadas análises e tratativas sobre a amostra de dados da planta solar escolhida.

Para uma melhor manipulação dos dados na fase de AED, foi utilizada a biblioteca *Pandas*, para auxiliar no processamento de matrizes e cálculos matemáticos utilizou-se a biblioteca *NumPy*, enquanto que para gerar os gráficos, foram utilizadas as bibliotecas *Matplotlib*, *SeaBorn* e *Plotly*. A manipulação de todas estas bibliotecas foi feita utilizando a linguagem *Python* através da plataforma *Google Colab*, uma plataforma na nuvem que permite reunir código com texto e imagens, em uma técnica conhecida como *notebook*.

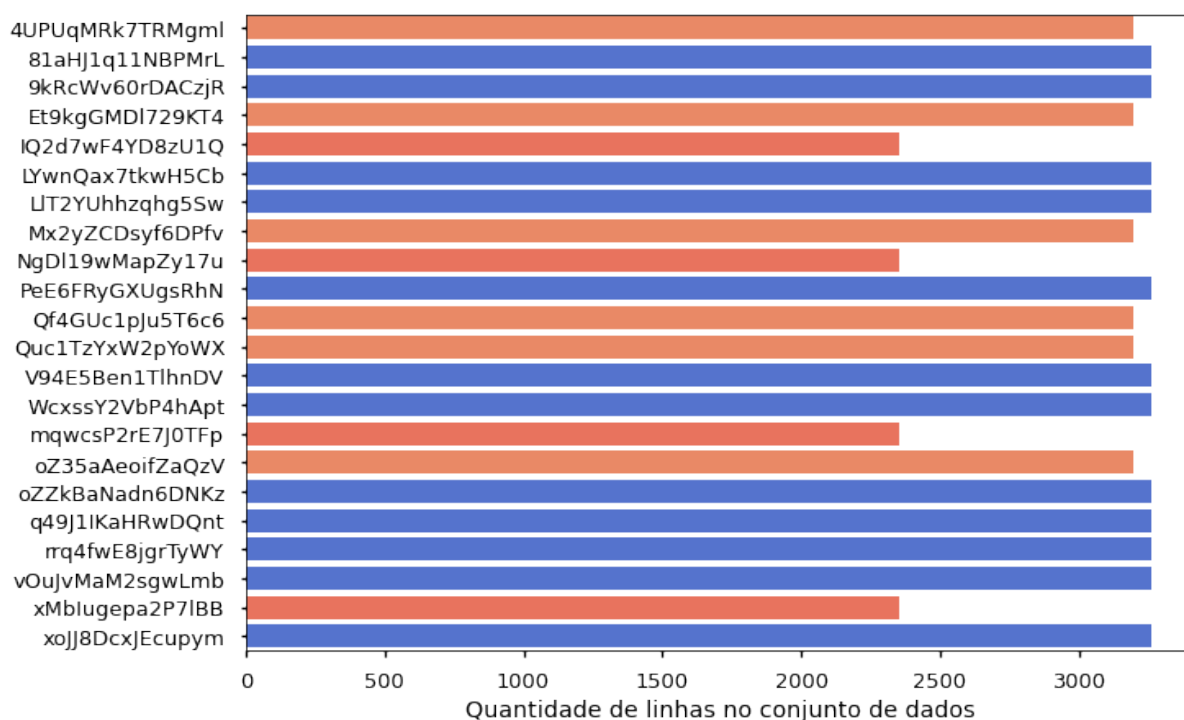
5.1.1 PREPARAÇÃO DOS DADOS

Primeiramente foi verificado se existe algum valor faltante (NaN) ou nulo no conjunto de dados. Percebeu-se uma abundante quantidade de valores iguais a zero para a geração de potência. Visto que o monitoramento continua durante o período da noite, isto faz sentido, no entanto, pode ser que o registro dos inversores tenha sido nulo mesmo durante o dia, o que pode indicar um mau funcionamento do sistema ou um problema na coleta dos dados.

Não foram encontrados valores faltantes na amostra, o que não é um indicativo de que não está faltando alguma linha com informação. Se sabe que as aferições foram realizadas a cada 15 minutos, e a coleta foi feita por um período de 24 horas durante 34 dias, para 22 inversores, portanto, a quantidade de linhas do conjunto de dados deveria ser igual a 71.808, o que não acontece, visto que a tabela de dados possui apenas 67.698 linhas com informação, o que significa que em algum momento não houveram registros dos dados.

Para um único inversor, deve haver 3.264 linhas de dados, portanto, a seguir será verificado a quantidade de informações gerada por cada inversor. Conforme se pode observar na Figura 15, onde cada inversor está representado por seu número de identificação ("SOURCE_KEY"), nenhum inversor gerou as 3.264 linhas de dados esperadas, no entanto, a maioria (em azul) gerou 3.259 linhas de dados, o que é muito próximo do esperado. Nota-se ainda que houve um padrão em relação ao registro dos dados, sendo 12 inversores com 3.259 linhas de informação, 6 com 3.195 linhas e 4 com 2.355 registros.

Figura 15 – Quantidade de linhas no conjunto de dados para cada inversor fotovoltaico (azul: 3.259 linhas de dados, laranja: 3.195 e vermelho: 2.355).



Fonte: (Autor, 2022)

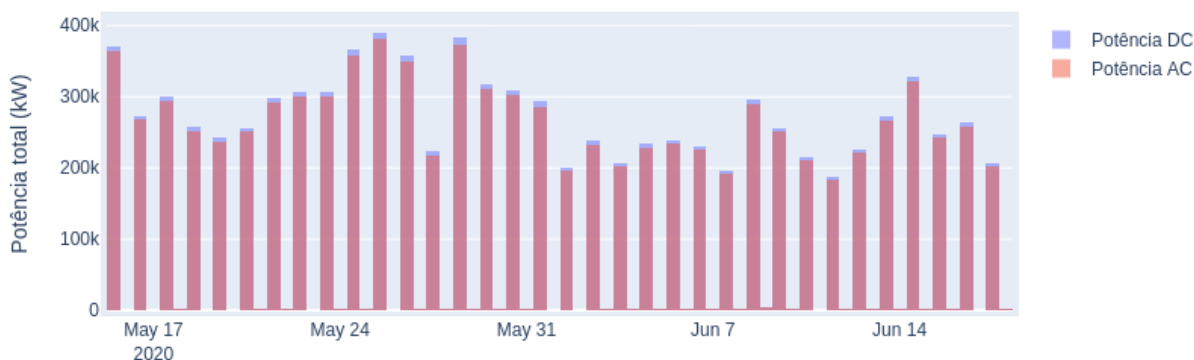
Para realizar a previsão de geração de potência do sistema fotovoltaico, foram considerados os 12 inversores que apresentaram uma maior quantidade de linhas de dados para os 34 dias. Com a eliminação dos registros dos demais 10 inversores, a amostra passou a ter 39.108 linhas com informação.

5.1.2 VISUALIZAÇÃO DOS DADOS

O gráfico da Figura 16 mostra a potência total AC² (vermelho) e DC (azul) gerada durante todos os 34 dias pelos 12 inversores selecionados, dividida em um intervalo de 6 horas para melhor visualização, sendo um período das 6h às 17h59 e outro das 18h às 5h59. Como esperado no período noturno (18h - 5h59) a potência gerada foi quase nula.

²No decorrer do trabalho serão utilizadas as siglas com os termos em inglês para se referenciar aos atributos de potência elétrica em corrente alternada e contínua, sendo elas, *Alternating Current (AC)* e *Direct Current (DC)*, corrente alternada e corrente contínua respectivamente.

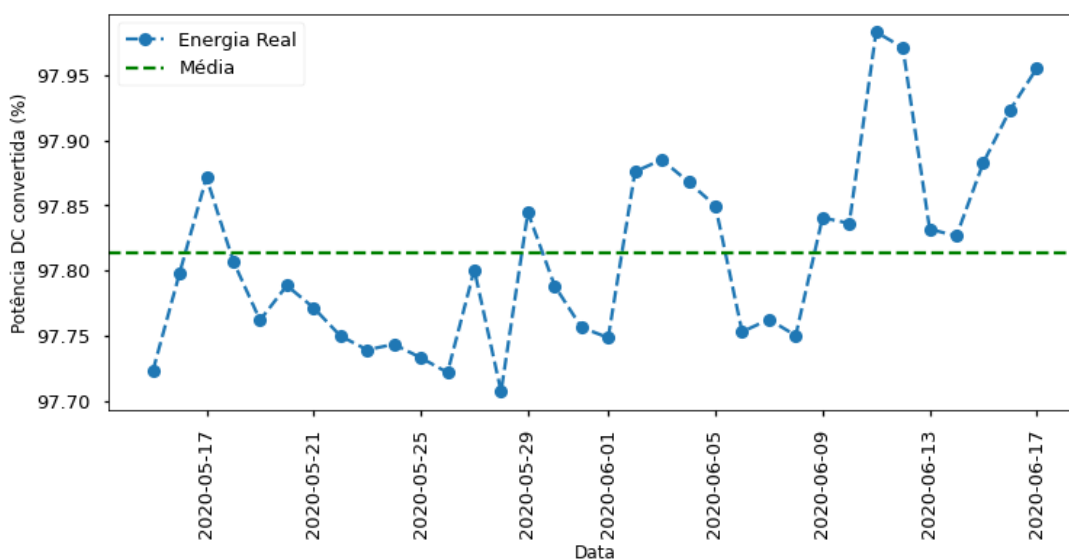
Figura 16 – Potência AC e DC total gerada em um intervalo de 6 horas para todos os 34 dias.



Fonte: (Autor, 2022)

O gráfico da Figura 16 evidencia que as perdas durante a conversão de potência foram mínimas, visto que os atributos quase que se sobrepuseram no gráfico. Para melhor visualização, o gráfico da Figura 17 mostra que a perda média no sistema fotovoltaico está em torno de 2,19%.

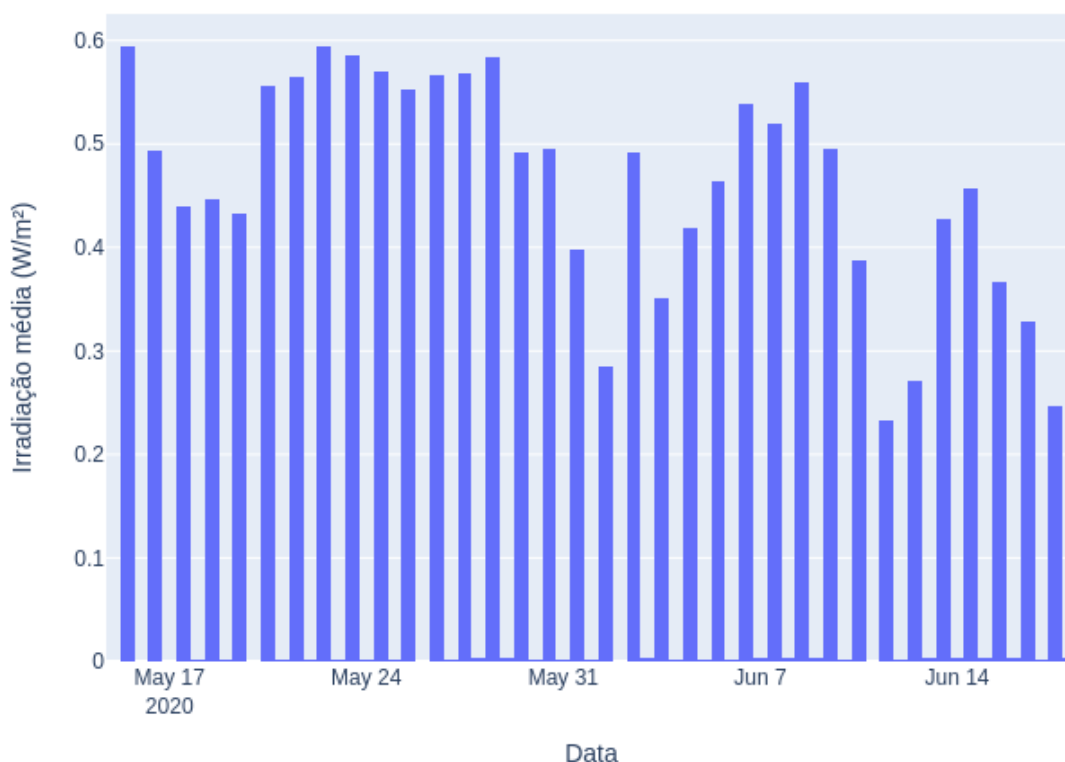
Figura 17 – % de potência DC convertida em potência AC.



Fonte: (Autor, 2022)

A irradiação registrada no mesmo período da geração de potência, apresentou um comportamento muito semelhante ao da potência gerada, conforme mostra a Figura 18.

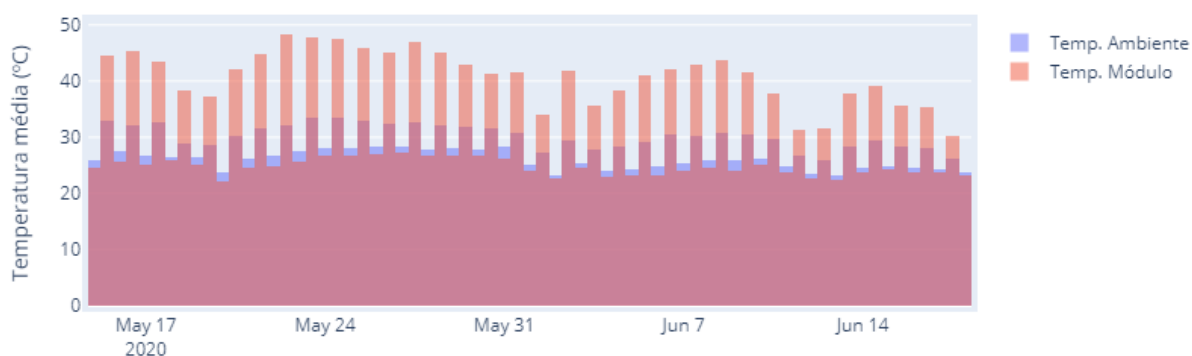
Figura 18 – Irradiação média em um intervalo de 6 horas para todos os 34 dias.



Fonte: (Autor, 2022)

Por fim, percebe-se que a temperatura do módulo fotovoltaico em azul segue o comportamento da temperatura ambiente em vermelho, como esperado (Figura 19).

Figura 19 – Temperatura média em um intervalo de 6 horas para todos os 34 dias.



Fonte: (Autor, 2022)

5.1.3 ANÁLISE DE ASSIMETRIA

Para uma simetria perfeita, espera-se valores para o coeficiente de assimetria de Pearson, conhecido em inglês como *skewness*, muito próximos de zero. Os atributos de geração de potência ("DC_POWER" e "AC_POWER") e irradiação ("IRRADIATION")

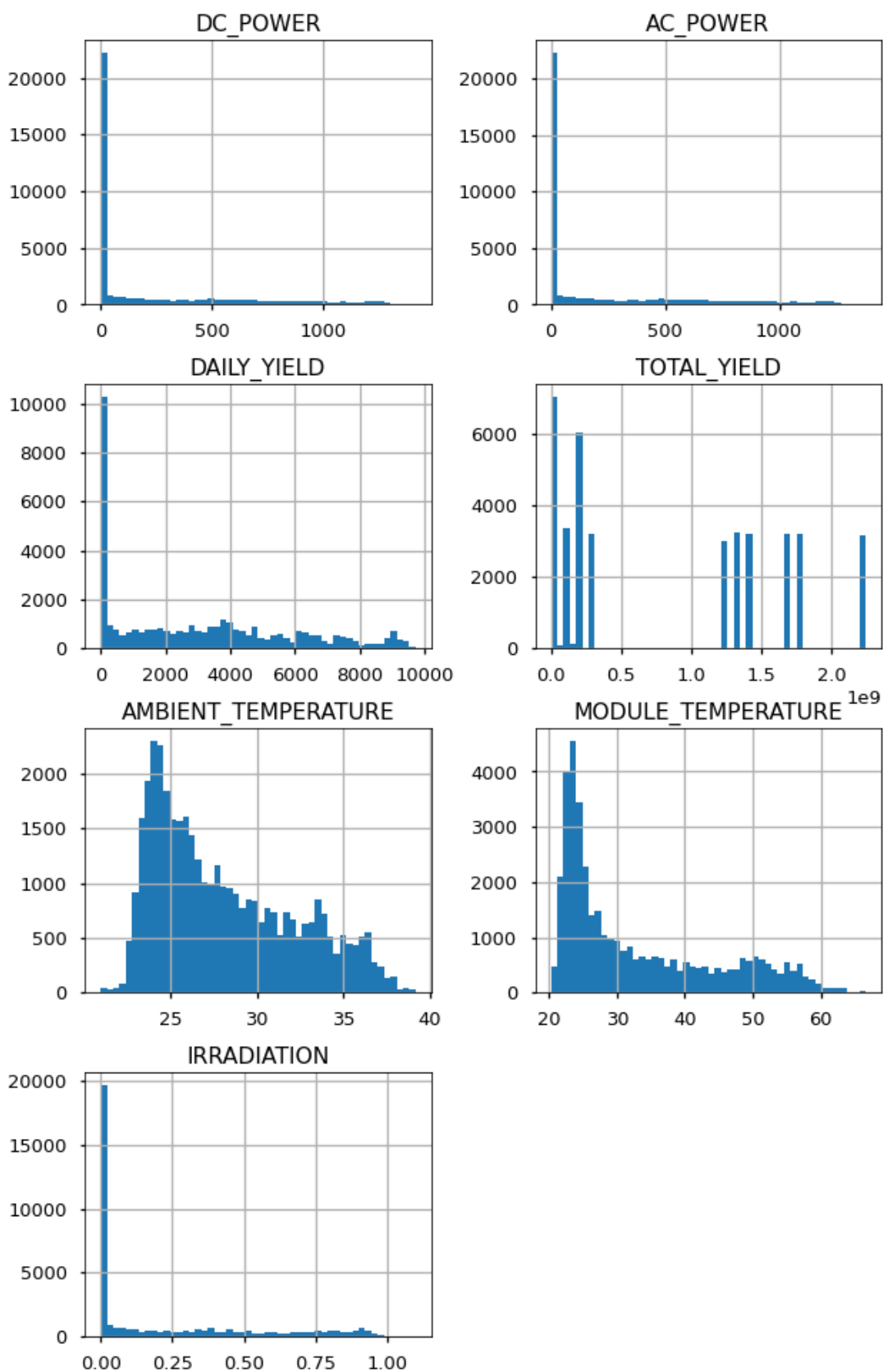
apresentam uma assimetria positiva relevante (Tabela 3), isto é evidenciado pela calda-longa (*tail-heavy*) apresentada pelos seus respectivos histogramas na Figura 20.

Tabela 3 – Coeficiente de assimetria de Pearson (*skewness*), antes de tratativas de valores nulos.

Atributo	Coeficiente de de assimetria de Pearson
DC_POWER	1,403455
AC_POWER	1,397904
IRRADIATION	1,061519
MODULE_TEMPERATURE	0,952207
AMBIENT_TEMPERATURE	0,648557
DAILY_YIELD	0,595715
TOTAL_YIELD	0,303067

Fonte: Autor, 2022

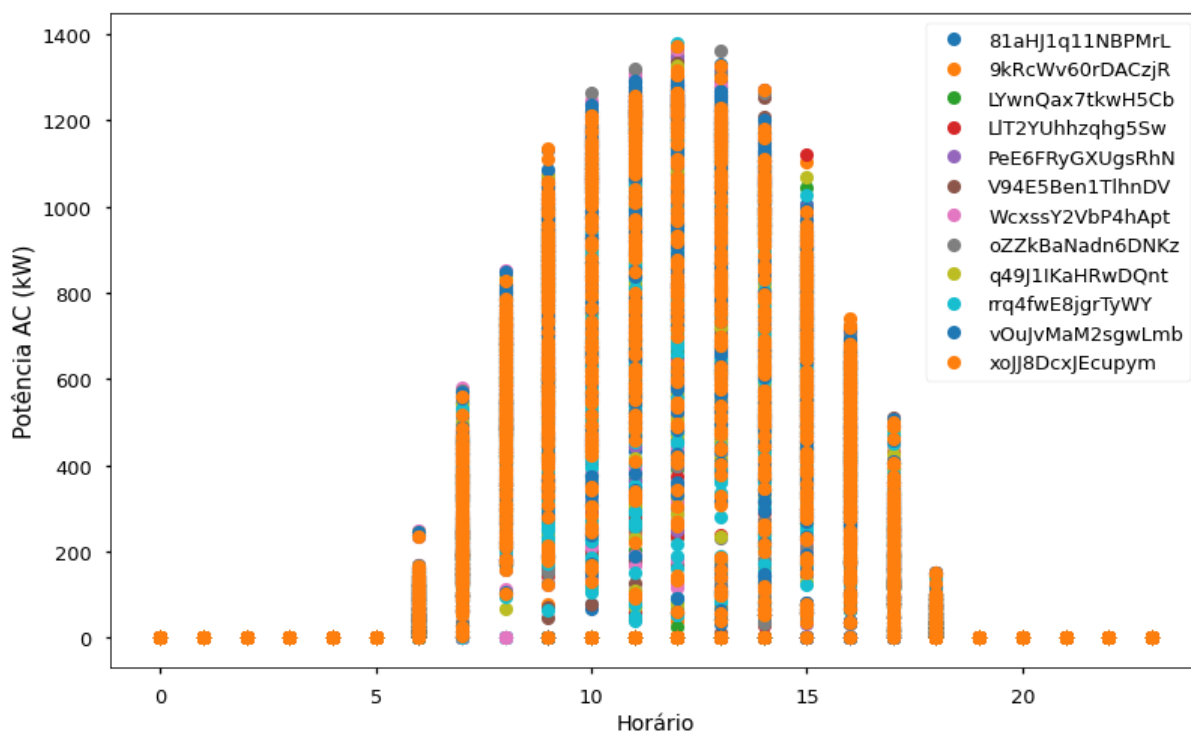
Figura 20 – Histograma com a simetria dos dados, antes da tratativa dos valores nulos.



Fonte: (Autor, 2022)

Percebe-se ainda que a calda-longa está no valor nulo, isto se dá, pois, durante a noite não há irradiação solar e conseqüentemente não se tem geração de potência elétrica. O gráfico da Figura 21, mostra que para todos os inversores, no período das 19h até 5h, o valor é nulo, sendo assim, este intervalo de dados pode ser descartado da análise.

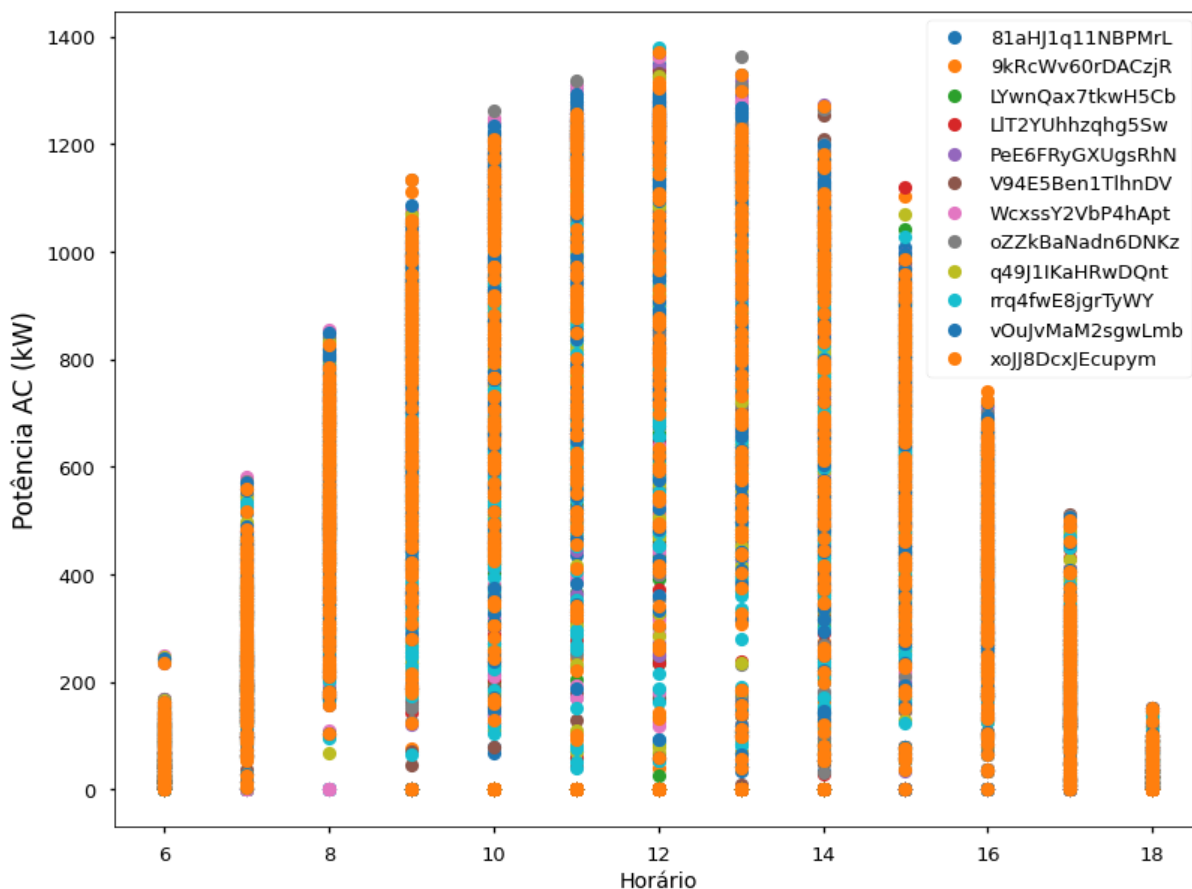
Figura 21 – Potência AC gerada durante o dia para todos os inversores.



Fonte: (Autor, 2022)

Desconsiderando os valores nulos, se tem o gráfico de geração de potência AC com o período ao qual o modelo será treinado, com informações referentes ao período das 6h às 18h45 (Figura 22).

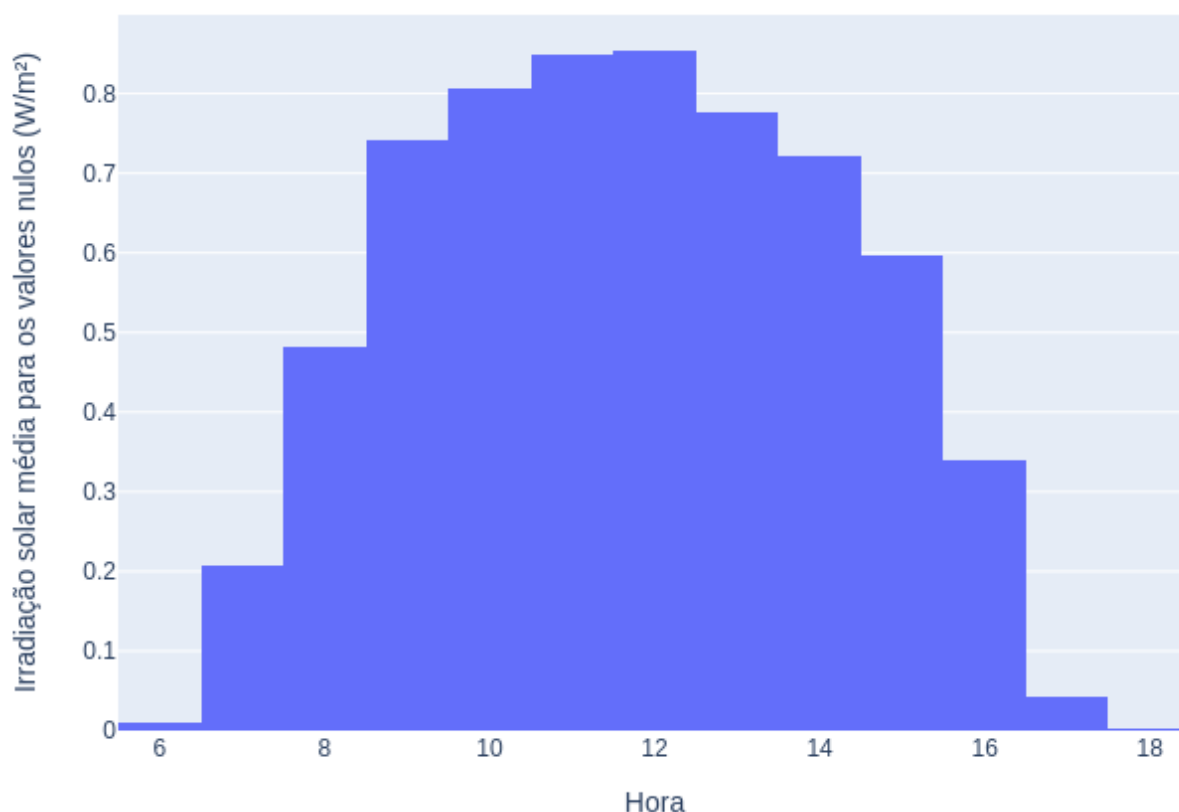
Figura 22 – Potência AC gerada durante o dia para todos os inversores (6h — 18h).



Fonte: (Autor, 2022)

Mesmo removendo o intervalo onde a geração de potência é nula, os histogramas dos dados ainda apresentaram uma calda-longa, o que pode indicar um mau funcionamento dos inversores. Para averiguar esta hipótese, foram reunidos os valores nulos registrados pelos inversores em uma amostra, para identificar o período em que estes ocorrem e se a irradiação possui influência sobre estes valores. Para a amostra dos dados com geração de potência AC nula, foram registrados os valores médios de irradiação apresentados na Figura 23.

Figura 23 – Irradiação média para quando a geração de potência foi nula.

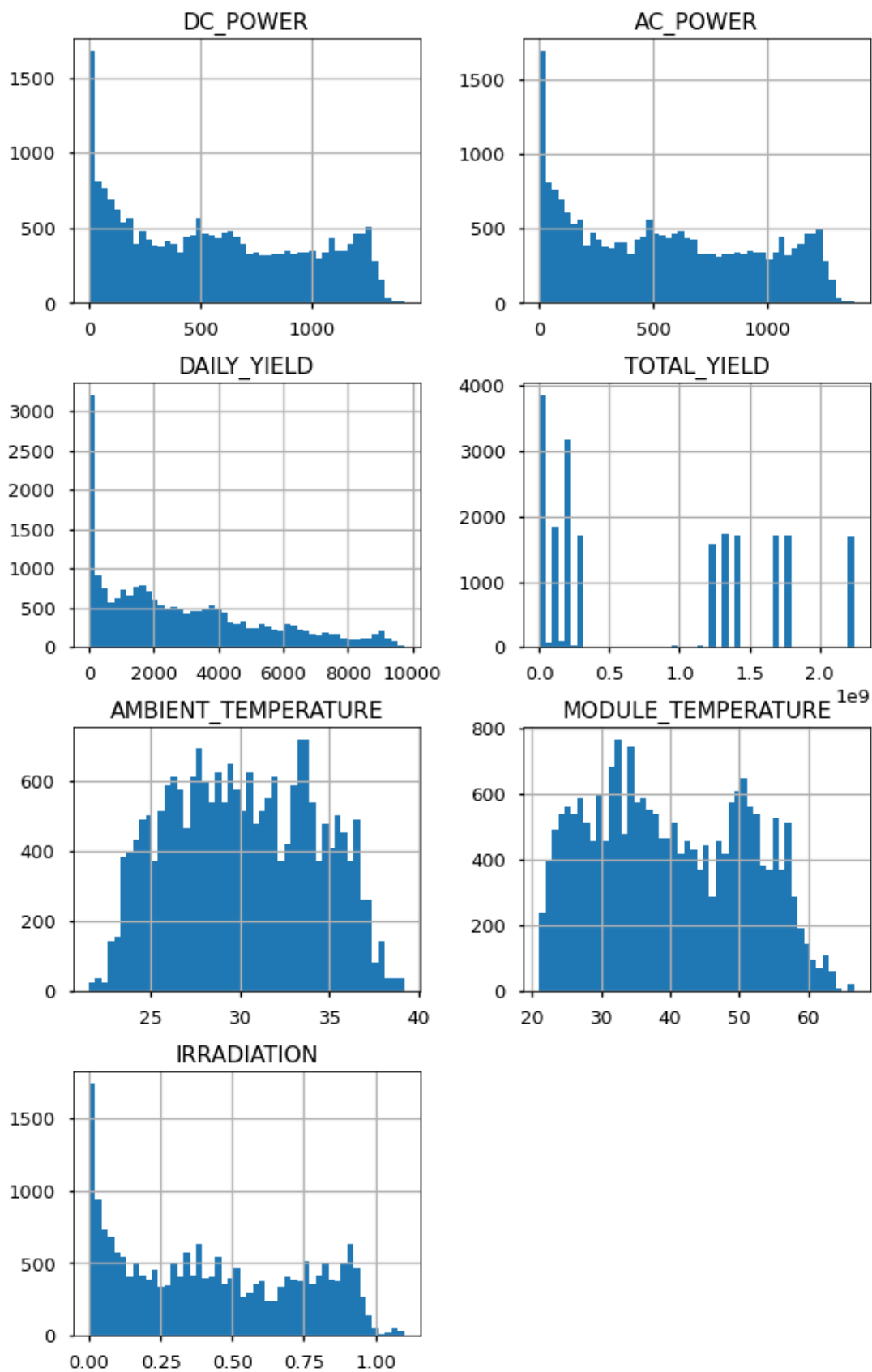


Fonte: (Autor, 2022)

Percebe-se que no período averiguado foi registrada irradiação suficiente para gerar potência, ou seja, os valores nulos de potência apresentados entre as 7h e 17h estão incorretos, o que pode indicar um mau funcionamento dos inversores, ou ter sido um período em que foi realizada alguma manutenção, precisando que o inversor fosse desligado, ou ainda, um problema no registro da informação. No Apêndice A está evidente o comportamento inesperado para alguns dos 12 inversores, onde ocorre uma queda abrupta da geração de potência elétrica. Visto que a irradiação apresentou um comportamento adequado durante o período, os valores nulos de potência elétrica neste intervalo (7h-17h) serão substituídos pelo seu valor antecedente, o que irá se aproximar do valor que deveria ter sido gerado naquele instante, visto que o passo do intervalo é de apenas 15 minutos. O resultado da nova amostra pode ser observado no Apêndice B.

Realizado o tratamento dos valores nulos, é perceptível na Figura 24 que a calda-longa que aparece nos atributos de potência, se assemelham com a calda-longa do atributo de irradiação, o que faz sentido, visto que quando a irradiação for igual a zero, a geração de potência elétrica tenderá a ser igual a zero.

Figura 24 – Histograma com a simetria dos dados, após tratativa dos valores nulos.



Fonte: (Autor, 2022)

Após as tratativas, os coeficientes de assimetria de Pearson comprovam que a amostra está com uma simetria consideravelmente melhor do que antes das tratativas.

Tabela 4 – Coeficiente de assimetria de Pearson (*skewness*), após tratativas de valores nulos.

Atributo	Coeficiente de de assimetria de Pearson
DAILY_YIELD	0,804027
TOTAL_YIELD	0,317565
DC_POWER	0,249611
AC_POWER	0,242521
IRRADIATION	0,211024
MODULE_TEMPERATURE	0,156068
AMBIENT_TEMPERATURE	0,055203

Fonte: Autor, 2022

Os atributos mais assimétricos do conjunto de dados são os de rendimento, isto se dá pelo seu comportamento, como o atributo "DAILY_YIELD" é uma variável acumulativa da potência gerada no dia, seu pico ocorre em um curto intervalo, por isso a característica decrescente no histograma. O "TOTAL_YIELD" apresenta uma constante para cada um dos 12 inversores, a alta volumetria de inversores com baixo índice de rendimento pode se dar pelo alto número de valores nulos registrados pela geração de potência elétrica, conforme pode ser observado no Apêndice A. Como foi realizada uma tratativa para os valores nulos, estes dois atributos já não representam o novo conjunto de dados, portanto é prudente removê-los do treinamento.

5.1.4 DEFINIÇÃO DO ATRIBUTO ALVO

A previsão da geração de potência elétrica do sistema fotovoltaico será realizada em relação a um atributo em um instante futuro no tempo. Caso fosse realizada a previsão do atributo de potência no instante presente, seria importante que os atributos de entrada não tivessem relação com o sistema fotovoltaico, pois caso contrário, iriam causar resultados tendenciosos. Sendo assim, o treinamento do modelo deveria ser feito, por exemplo, apenas com dados sazonais e meteorológicos.

No presente trabalho, será feita a previsão da potência com base em atributos de entrada que também advêm do próprio sistema fotovoltaico, logo, para evitar uma tendência na previsão, o treinamento será realizado para prever a geração de potência AC em um instante futuro ("AC_POWER_FUTURO"). Portanto, o atributo alvo continua sendo o "AC_POWER", mas deslocado em 15 minutos a frente no tempo.

5.1.5 ANÁLISE DE CORRELAÇÃO

Antes de avaliar a correlação das variáveis, foram criados mais dois atributos. Visto que dados de irradiação dependem da sazonalidade, foram criadas as colunas "HOUR" (hora) e "DAY" (dia), de modo a verificar se estes atributos possuem correlação com o atributo alvo, a potência AC gerada. Decidiu-se não criar o atributo relacionado ao mês, dado que o intervalo de registro dos dados ocorre entre os dias de apenas dois meses, o que não resultaria em uma representatividade sazonal satisfatória.

Utilizando a função de correlação da biblioteca *Keras*, obteve-se o resultado apresentado na Tabela 5, onde cada índice representa o quanto que o atributo influencia no atributo alvo ("AC_POWER_FUTURO"), quanto mais próximo de 1, maior é a correlação entre os atributos.

Tabela 5 – Correlação em relação ao atributo alvo.

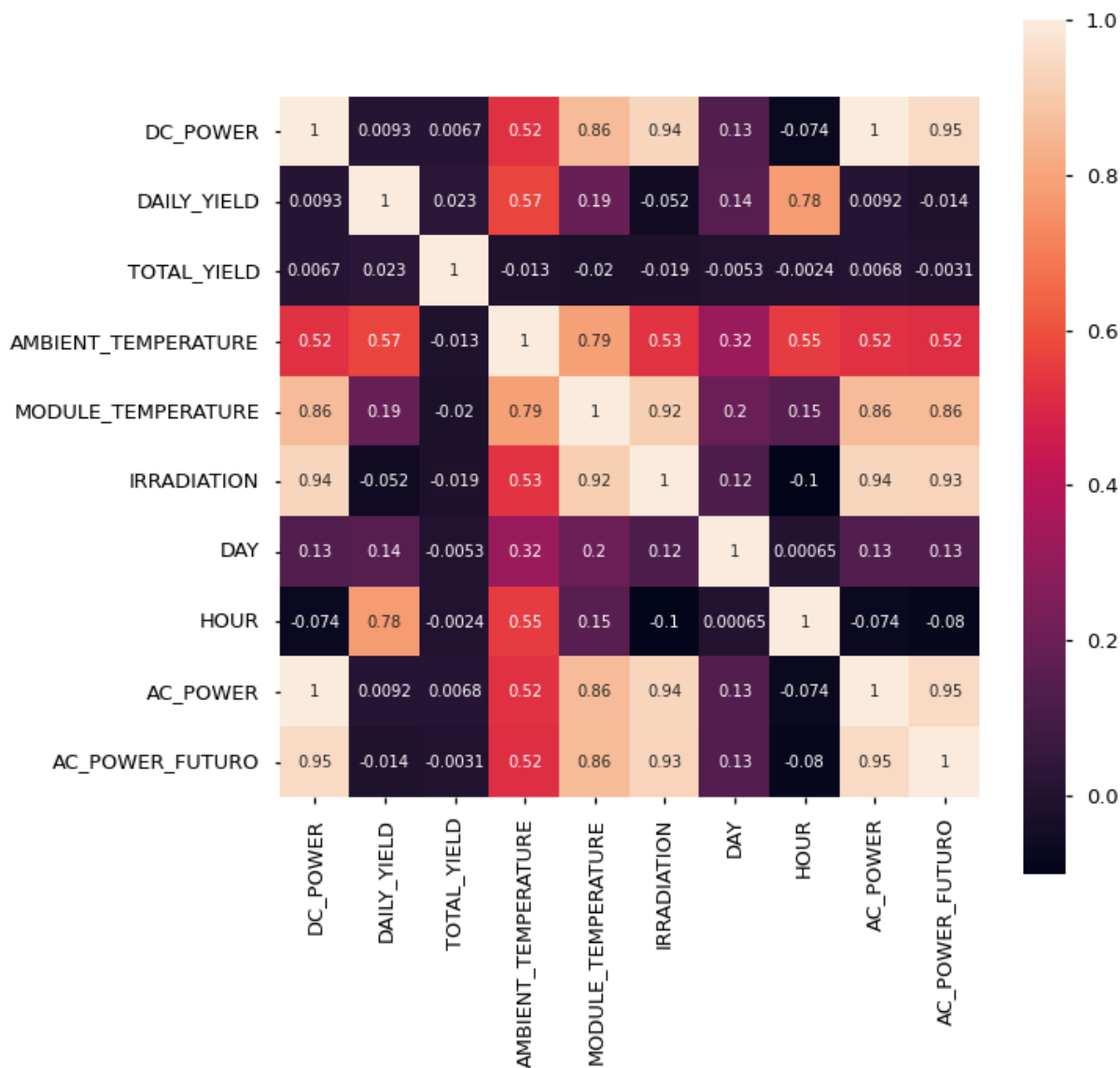
Atributo	Índice de Correlação
DC_POWER	0,952057
AC_POWER	0,952038
IRRADIATION	0,932355
MODULE_TEMPERATURE	0,859518
AMBIENT_TEMPERATURE	0,516717
DAY	0,134700
HOUR	-0,080288

Fonte: Autor, 2022

Foi vista anteriormente a perda baixíssima na conversão da potência DC para a AC, logo, presume-se que a correlação entre estes dois atributos seja muito alta, como de fato o é. Interessante constatar que o atributo "DC_POWER" possui uma correlação aproximadamente igual ao atributo "AC_POWER". Percebe-se ainda uma forte correlação positiva na irradiação e na temperatura do módulo, o que faz sentido, visto que quanto maior a irradiação solar, maior deve ser a potência gerada, e quanto maior a irradiação sobre os módulos, maior será sua temperatura. Os atributos de tempo, mostram que não existe uma correlação linear com a geração de potência, visto que os coeficientes são muito próximos de zero, isto pode se dar pelo período curto dos dados da amostra, não evidenciando uma sazonalidade nos dados.

Da Figura 25, percebe-se que a correlação entre os atributos de temperatura são muito próximas de 1, o que pode ser um forte indício de multicolinearidade entre estes atributos. Fica evidente que o atributo de potência DC apresenta multicolinearidade com o atributo de potência AC, assim como já havia ficado claro através da Figura 17. Dado que o atributo "DC_POWER" representa a mesma informação que o atributo "AC_POWER", este não será utilizado para o treinamento do modelo.

Figura 25 – Índices de correlação entre os atributos.



Fonte: (Autor, 2022)

Dada a alta correlação com a variável de saída, o atributo "AC_POWER" deveria ser excluído da amostra caso a previsão da geração de potência elétrica fosse uma previsão do instante presente. No entanto, como a previsão se dará em relação ao valor "AC_POWER" em um instante futuro no tempo, esta variável pode ser considerada no conjunto de treinamento.

5.1.6 ANÁLISE DE MULTICOLINEARIDADE

Os atributos preditores são os atributos que serão considerados na entrada do modelo para se prever o atributo de saída. Na multicolinearidade, atributos preditores que se correlacionam com outros preditores, são atributos redundantes na amostra

de dados. Com a análise de correlação, já se pôde constatar alguns atributos com multicolinearidade, aqui será realizado uma análise mais minuciosa.

Os atributos qualitativos ("DATE_TIME" e "SOURCE_KEY") e o atributo dependente ("AC_POWER_FUTURO"), serão removidos da análise de multicolinearidade, visto que não serão considerados para a entrada do modelo preditivo.

Na Tabela 6 estão reunidos os valores VIF para cada atributo independente, ou preditor. Percebe-se um alto valor para os atributos de temperatura e irradiação, principalmente para os de temperatura, por este motivo, primeiramente foram removidos os atributos "AMBIENT_TEMPERATURE" e "MODULE_TEMPERATURE".

Tabela 6 – Fator de inflação de variância para os atributos independentes.

Atributo	VIF
AMBIENT_TEMPERATURE	226,559285
MODULE_TEMPERATURE	368,071720
IRRADIATION	66,457431
AC_POWER	23,743236
HOUR	21,334049
DAY	5,396682

Fonte: Autor, 2022

Em seguida o VIF foi novamente calculado, excluindo como feito anteriormente valores acima de 5, este procedimento foi realizado até se chegar em um valor considerado satisfatório e homogêneo entre as variáveis. Para o caso em análise, na situação considerada sem colinearidade entre os atributos independentes, restaram apenas os atributos "HOUR", "DAY" e "IRRADIATION", conforme Tabela 7.

Tabela 7 – Melhor índice de fator de inflação de variância para os atributos independentes.

Atributo	VIF
HOUR	3,935756
DAY	3,827207
IRRADIATION	2,603135

Fonte: Autor, 2022

Com base no resultado obtido para a análise de multicolinearidade, conclui-se que os atributos da Tabela 7 não possuem colinearidade entre si. Como foi visto na análise de colinearidade para com o atributo alvo ("AC_POWER"), os atributos de tempo ("HOUR" e "DAY") apresentaram um índice de colinearidade muito baixo, por este motivo, estima-se que se obterá um resultado satisfatório de treinamento na regressão linear utilizando somente a variável "IRRADIATION" como entrada para o modelo. Para confirmar esta decisão, a seguir foi gerado um relatório utilizando o

modelo *Ordinary Least Square* (OLS) com o método *Least Squares*, considerando algumas combinações de atributos de entrada, a título de comparação.

5.1.7 ANÁLISE DE REGRESSÃO LINEAR

Utilizando a biblioteca *ScikitLearn* foi gerado um relatório utilizando o modelo *Ordinary Least Square* (OLS) com o método *Least Squares*. Na Tabela 8 estão reunidos os valores obtidos para a amostra considerando todos os atributos como dados de entrada (Caso 1), utilizando os atributos selecionados pela avaliação do índice de fator de inflação de variância (Caso 2) e os valores registrados considerando somente o atributo "IRRADIATION" como entrada (Caso 3).

Tabela 8 – Relatório OLS.

Atributo	Valor t (Caso 1)	Valor t (Caso 2)	Valor t (Caso 3)
IRRADIATION	37,044	371,007	375,171
MODULE_TEMPERATURE	-2,287	-	-
AMBIENT_TEMPERATURE	3,279	-	-
AC_POWER	114,915	-	-
DAY	2,594	9,271	-
HOUR	-1,864	5,649	-
R ²	0,920	0,870	0,869
R ² Ajustado	0,920	0,870	0,869

Fonte: Autor, 2022

Para o Caso 1 o modelo explica 92,0% da variância dos dados, para o Caso 2 explica 87,0%, enquanto que para o Caso 3 o modelo consegue explicar 86,9% da variância. O maior valor para o primeiro caso se deve principalmente ao atributo "AC_POWER", que apresentou o melhor valor t quando comparado com os demais, tendo, portanto, a maior relevância para o modelo preditivo. Retirando o "AC_POWER" da entrada, o desempenho do modelo para explicar a variância dos dados cai para 87,0%, um R² equivalente ao apresentado pelo Caso 2.

Um modelo considerado satisfatório para a previsão do atributo alvo, possui um R² com valor próximo de 1. Comparando os três casos, percebe-se que o caso com todos os atributos (Caso 1) apresenta um R² melhor. É importante que o R² e o R² Ajustado possuam um valor próximo, que é o que acontece em todos os casos, isto indica que em geral os atributos considerados como entrada possuem relevância.

Percebe-se ainda que os atributos "DAY", "HOUR", "MODULE_TEMPERATURE" e "AMBIENT_TEMPERATURE" apresentaram um valor consideravelmente baixo para o valor t, o que indica uma baixa relevância para o modelo preditivo. O valor t, ou estatística t, indica a capacidade preditiva de cada atributo individualmente.

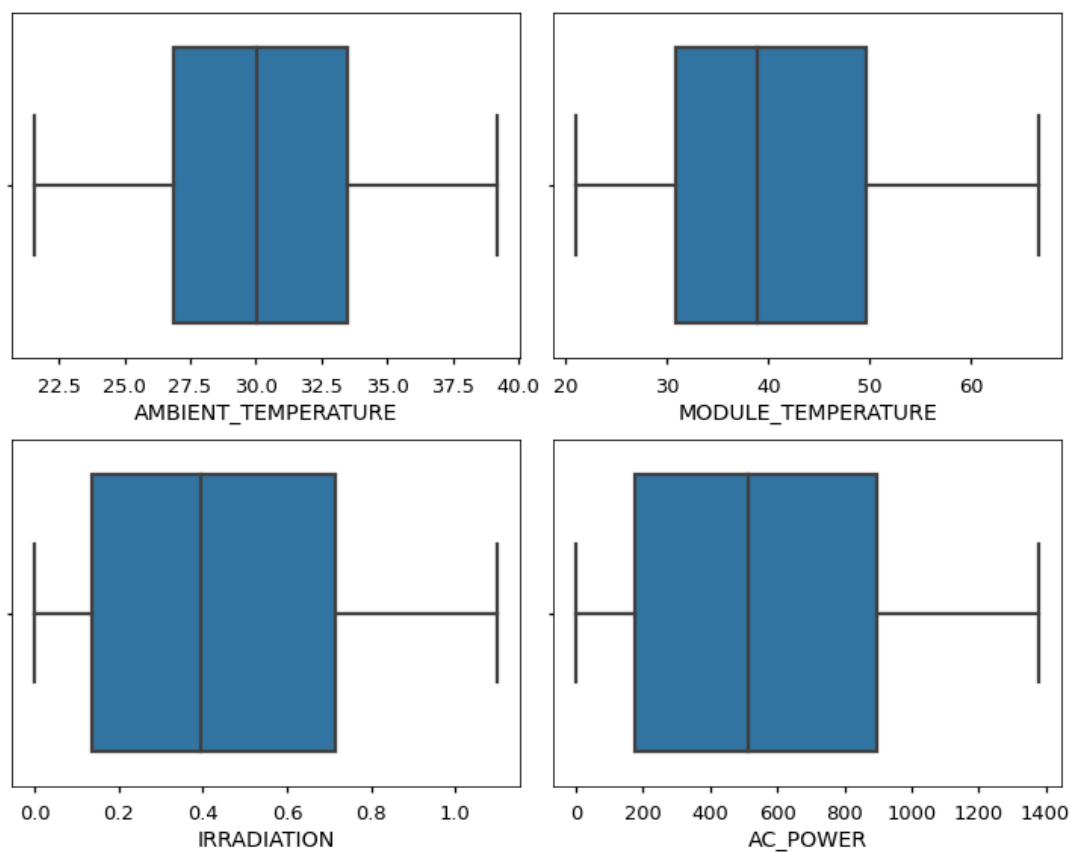
O modelo de regressão linear será treinado com um único atributo de entrada ("IRRADIATION") e com todos os atributos de entrada, a título de comparação. No entanto, com a análise feita até então, pode-se ter uma ideia de como serão os resultados do treinamento.

Importante ressaltar que esta análise é válida para os modelos lineares. Para a rede neural, um grande número de atributos pode ser interessante, visto que esta transforma os atributos através de combinações não lineares, criando atributos mais discriminativos.

5.1.8 ANÁLISE DE QUARTIL

Por fim, foi verificado se os atributos, com exceção dos atributos de tempo, possuem valores muito acima da média (*outliers*) na amostra de dados, situações estas que podem influenciar negativamente o treinamento do modelo, além disso, podem indicar um problema na geração de energia elétrica da planta fotovoltaica. Nos diagramas de caixa da Figura 26, percebe-se que não houveram *outliers* nos atributos durante o período em que foram registradas as informações, visto que não existem pontos discrepantes além do limite superior ou inferior.

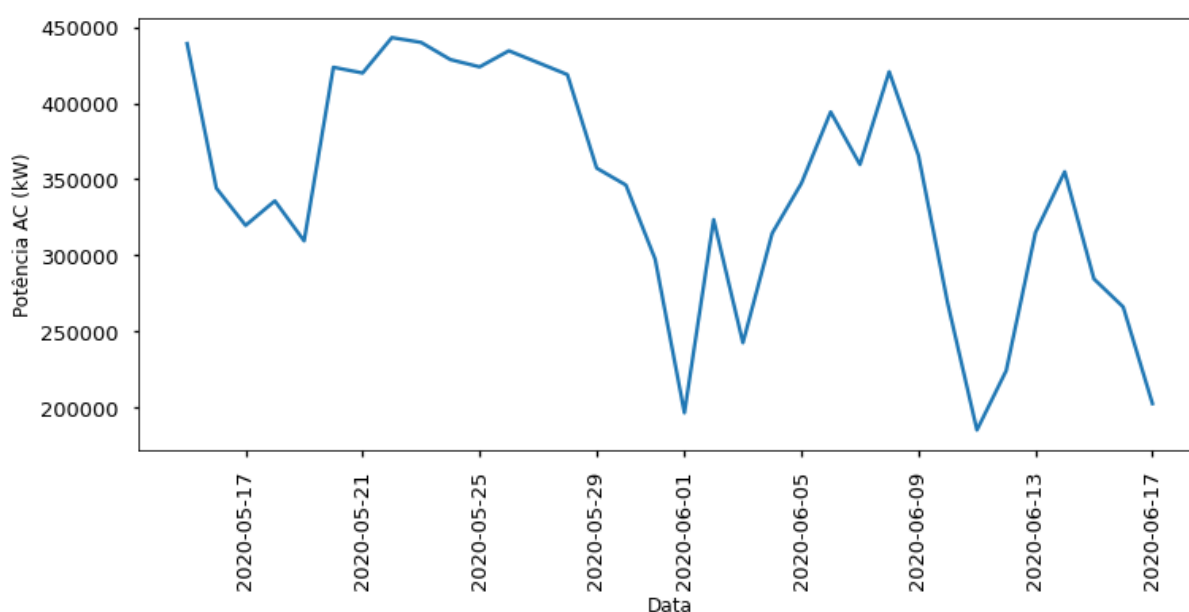
Figura 26 – Diagrama de caixa dos atributos.



5.2 MODELAGEM

Com a etapa de AED finalizada, foi iniciada a etapa de modelagem. A previsão da geração de potência elétrica da planta fotovoltaica foi feita com dois modelos lineares (Regressão Linear e Regressão Polinomial) e dois modelos de redes neurais (MLP e LSTM). A Figura 27 mostra o somatório da potência AC gerada pela planta solar durante todo o período de 34 dias e que deverá ser previsto pelos modelos. O *framework* foi desenvolvido na linguagem *Python*, utilizando a plataforma *Google Colab*.

Figura 27 – Potência AC gerada pela planta solar durante os 34 dias.



Fonte: (Autor, 2022)

Para realizar a modelagem de regressão foi utilizada a biblioteca *Scikit-Learn*. Para o treinamento dos dados com rede neural, foi utilizada a popular API *Keras*, uma API de alto nível, projetada para criar, treinar, avaliar e implementar redes neurais.

5.2.1 DIVISÃO DA AMOSTRA

A primeira etapa e que serviu para todos os modelos, foi dividir o conjunto de dados em treino e teste. Para treinar o modelo foram utilizados os dados referentes aos primeiros 27 dias, enquanto para testar o modelo, foram reservados os últimos 7 dias. Para os modelos de rede neural, 15% do conjunto de treinamento foi reservado para avaliar o modelo, ou seja, 15% do conjunto de dados não será utilizado no treinamento, mas para verificar o quão bem o modelo consegue prever dados futuros, o que auxiliou na escolha dos hiperparâmetros do modelo. Sendo assim, se tem reservado da amostra total, aproximadamente 79,37% para a amostra de treino (16.800 linhas de dados) e 20,63% para a amostra de teste (4.367 linhas de dados).

Antes do treinamento dos modelos, os dados foram normalizados em uma escala de 0 a 1, a normalização é importante visto a divergência na magnitude dos atributos, atributos com valores maiores iriam causar tendenciosidades no modelo. Tanto a amostra de teste quanto a amostra completa, foram normalizadas com os dados de treinamento, visto que normalizar o conjunto de teste com o conjunto de dados completo iria carregar informações da amostra completa para o conjunto de teste, influenciando o resultado final.

Para o cálculo de perda, por se tratar de um problema de regressão e não haver *outliers* na amostra completa, foi adotada a função MSE. Foi calculada também a sua raiz quadrática de modo a obter o RMSE do modelo, auxiliando no entendimento da variância do resultado obtido pelas previsões.

5.2.2 MODELOS LINEARES

Como já supracitado, serão utilizados dois modelos lineares, o de regressão linear e o de regressão polinomial. O modelo de regressão linear simples recebe somente um único atributo como entrada, referente ao Caso 3, para utilizar o Caso 1, com múltiplos atributos de entrada, foi aplicado o modelo de regressão com múltiplas variáveis e ainda o modelo de regressão polinomial.

5.2.2.1 MODELO REGRESSÃO LINEAR

Em conformidade com os procedimentos utilizados para a construção dos modelos de redes neurais, a normalização dos dados também foi realizada aqui. Dessa forma, utilizou-se a entrada normalizada no treinamento. Se tratando de uma regressão linear simples, foi utilizado na entrada do modelo um único atributo, o de irradiação, conforme Caso 3 apresentado na Subseção 5.1.7.

Utilizando o valor da irradiação, o modelo irá tentar aproximar uma reta que consiga, de melhor forma possível, prever o atributo de saída (Potência AC).

5.2.2.2 MODELO REGRESSÃO LINEAR COM MULTIVARIÁVEIS

No modelo de regressão linear com múltiplas variáveis, como o próprio nome evidência, será utilizado mais do que um único atributo de entrada, aqui, foram utilizados para a entrada do modelo todos os seis atributos do Caso 1 da Subseção 5.1.7, de modo a se tentar prever o valor de potência AC futuro.

Diferente do modelo de regressão linear simples, com uma única variável, neste caso como são utilizados vários atributos na entrada, o modelo irá realizar uma combinação linear com cada um dos atributos, considerando a contribuição de cada um deles, resultando não em uma simples reta, mas sim um hiperplano.

5.2.2.3 MODELO REGRESSÃO POLINOMIAL COM MULTIVARIÁVEIS

Para o treinamento foi utilizado uma regressão polinomial de ordem 2. Assim como foi feito na regressão linear, o modelo de regressão polinomial foi treinado considerando todos os seis atributos de entrada. Neste caso, para uma entrada de seis atributos, o modelo realiza uma transformação para doze atributos, dada a característica polinomial do modelo, realizando uma combinação linear com todas as variáveis.

5.2.3 MODELOS DE REDE NEURAL

Em um modelo de rede neural é necessário definir um otimizador, algoritmo que utiliza o método de *Backpropagation Through Time* (Retropropagação Através do Tempo), método de otimização baseado no algoritmo de descida do gradiente. Tanto para a rede MLP quanto para a LSTM, foi utilizado o otimizador *Adam*, visto que é um método simples de implementar, computacionalmente eficiente e necessita de pouca memória (KINGMA; BA, 2017).

Para o algoritmo de retropropagação é necessário que sejam definidas as épocas e o tamanho do lote (*batch size*) de cada época. A cada época, o algoritmo passa por todas as amostras dos dados de treino, atualizando os pesos da rede, quando em uma iteração o treinamento passa por todas as amostras, uma época é concluída. O tamanho do lote irá definir quantos lotes, subconjunto dos dados de treinamento, serão utilizados a cada iteração (BROWNLEE, 2017). A título de exemplo do que foi explicado, supondo que um modelo de RNN possua 1.000 amostras e um tamanho de lote igual a 100, durante o treinamento a amostra será dividida em 10 (1.000/100) subconjuntos, sendo assim, uma época é finalizada quando o modelo treinar os dados passando por 10 subconjuntos, cada um contendo 100 amostras.

Para o treino de rede neural, foram utilizadas duas metodologias, tanto para a rede MLP, quanto para a rede LSTM, sendo elas o método padrão, também conhecido como método *Vanilla*, e o método de janela, também conhecido como o método *Window*.

5.2.3.1 MÉTODO VANILLA

No método *vanilla*, ou método padrão, para cada iteração que o modelo realizar, será considerado o valor de entrada anterior. Por exemplo, supondo que existem os valores de irradiação (A, B e C) que correspondem aos valores de potência elétrica (A', B' e C'), na primeira iteração do treinamento, o modelo irá tentar prever B' olhando para A, em seguida tentará prever C' olhando para B e assim sucessivamente.

5.2.3.2 MÉTODO WINDOW

No método *Window*, ou método de janela, a única diferença para o método padrão é que a cada iteração o treinamento irá olhar para um intervalo de dados

anteriores, para uma janela de informações passadas. Anteriormente, ao tentar prever um valor de saída, olhava-se para um único valor passado, agora, serão considerados n valores passados.

Para definir o tamanho da janela, considerou-se que o modelo deverá olhar para um conjunto de dados que represente o período de 1 hora. Sendo 12 inversores na amostra, a cada 1 hora, são obtidas 48 linhas de dados, 12 linhas de dados a cada 15 minutos. Portanto, para um *look back*³ igual a 48, a cada iteração é levado em consideração os dados da última hora para realizar uma previsão.

5.2.3.3 MODELO MLP

Na Tabela 9 estão reunidos os hiperparâmetros utilizados para treinar o modelo MLP padrão, estes foram os que apresentaram um desempenho satisfatório durante a etapa de validação. Foi considerada uma única camada de entrada, uma camada oculta com 64 neurônios e uma camada de saída com um único neurônio.

Como função de ativação utilizou-se a sigmoide. O treino foi realizado em 11 épocas com um tamanho do lote (*batch size*) igual a 32 para o modelo com um único atributo de entrada, e 36 épocas para o modelo com múltiplos atributos.

Tabela 9 – Hiperparâmetros utilizados no modelo MLP padrão.

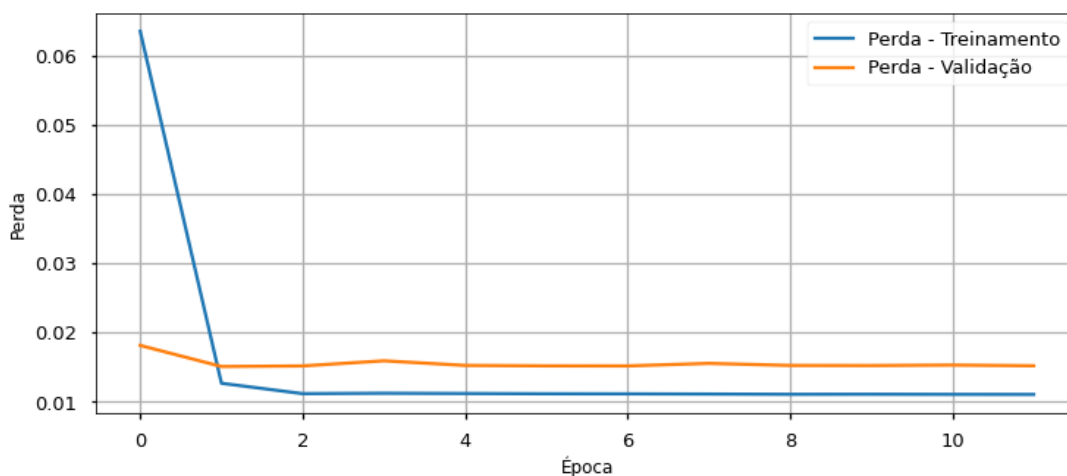
Atributos	Neurônios	Épocas	Tamanho do lote	Ativação	Otimizador
Univariável	64	11	32	Sigmoide	Adam
Multivariável	64	36	32	Sigmoide	Adam

Fonte: Autor, 2022

A Figura 28 mostra o histórico de perdas durante toda a etapa de treinamento. Aqui a perda é o MSE, a cada época o algoritmo calcula o erro quadrático médio, se após 10 épocas o erro não diminuir, o treinamento é interrompido. Percebe-se que ao atingir 11 épocas as perdas se estabilizam, por este motivo, 11 épocas são suficientes para treinar o modelo, enquanto para múltiplas variáveis são necessárias 36 épocas, conforme Figura 29.

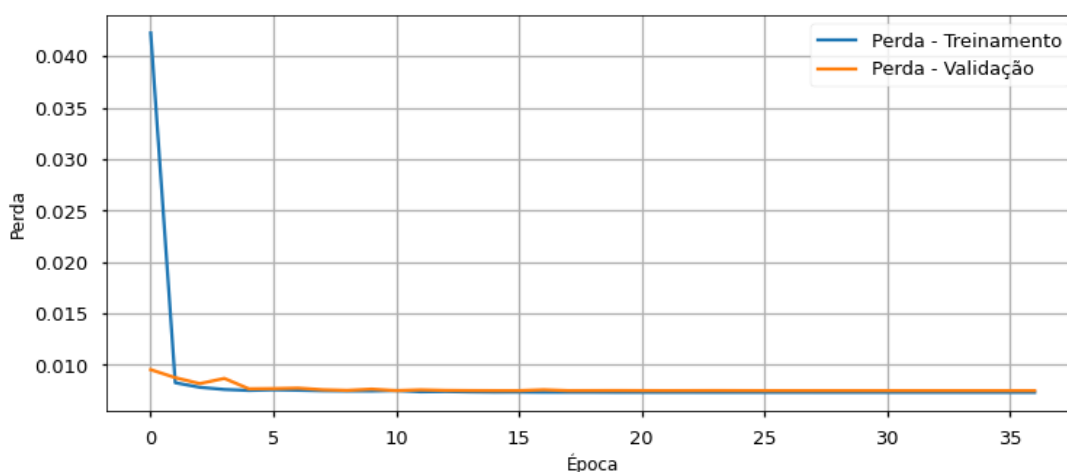
³*look back* é uma variável utilizada no algoritmo desenvolvido e que corresponde a quantidade de passos anteriores que o treinamento irá considerar, ou seja, quantidade de atributos passados que serão considerados para a previsão do atributo futuro.

Figura 28 – Perdas (MSE) para MLP com uma única variável de entrada.



Fonte: (Autor, 2022)

Figura 29 – Perdas (MSE) para MLP com múltiplas variáveis de entrada



Fonte: (Autor, 2022)

5.2.3.4 MODELO MLP COM MÉTODO WINDOW

Os hiperparâmetros utilizados para a rede MLP com método de janela foram os mesmos que os utilizados para o método padrão, conforme Tabela 10.

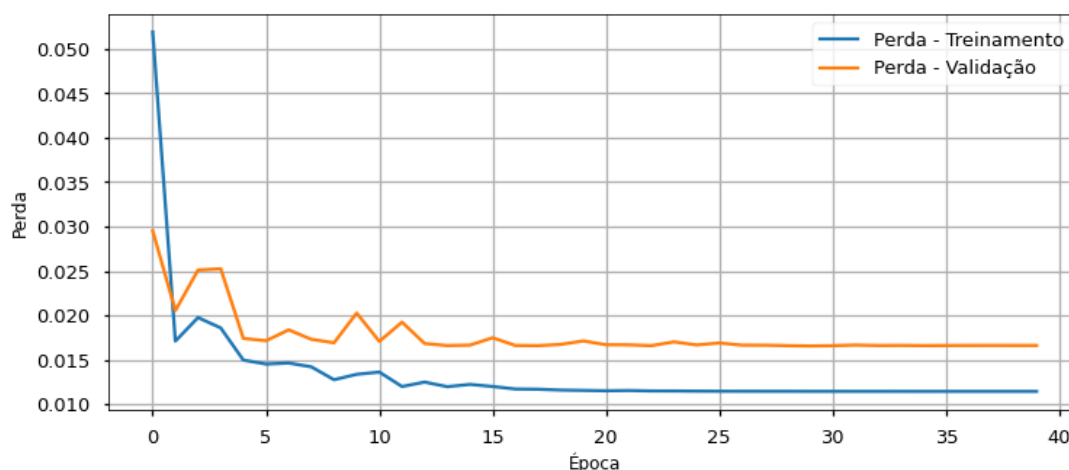
Tabela 10 – Hiperparâmetros utilizados no modelo MLP com método *Window*.

Atributos	Neurônios	Épocas	Tamanho do lote	Ativação	Otimizador
Univariável	64	39	32	Sigmoide	Adam
Multivariável	64	52	32	Sigmoide	Adam

Fonte: Autor, 2022

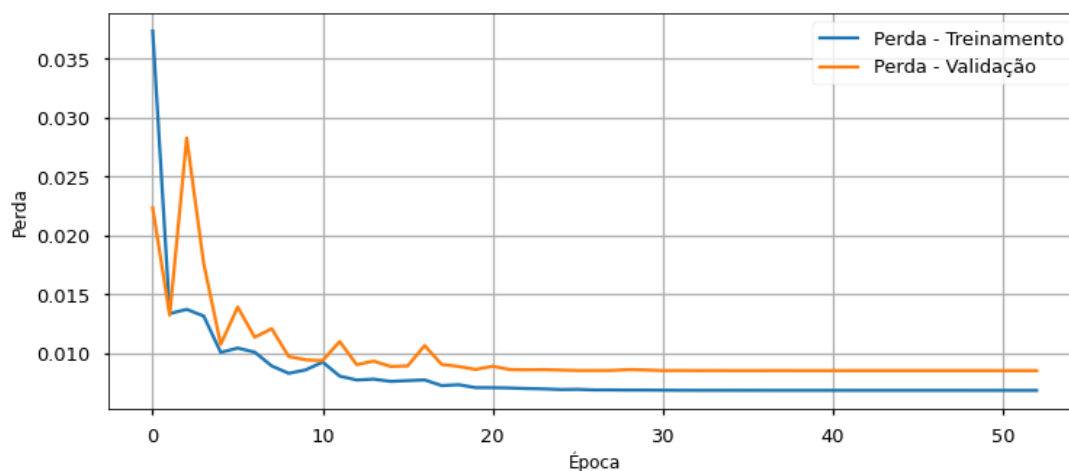
A Figura 30 mostra o histórico de perdas durante toda a etapa de treinamento, percebe-se que para atingir o equilíbrio foram necessárias mais épocas do que o método padrão, ao atingir 39 épocas as perdas se estabilizaram. Para múltiplas variáveis, a estabilização ocorreu em 52 épocas (Figura 31).

Figura 30 – Perdas (MSE) para MLP com método *Window* e uma única variável de entrada.



Fonte: (Autor, 2022)

Figura 31 – Perdas (MSE) para MLP com método *Window* e múltiplas variáveis de entrada.



Fonte: (Autor, 2022)

5.2.3.5 MODELO LSTM

Diferente da MLP a rede LSTM espera que os dados de entrada sejam fornecidos com uma estrutura de matriz na forma de: [amostras, etapas de tempo, características]. Para título de comparação, os hiperparâmetros utilizados para a LSTM foram os mesmos que os utilizados na rede MLP (Tabela 11).

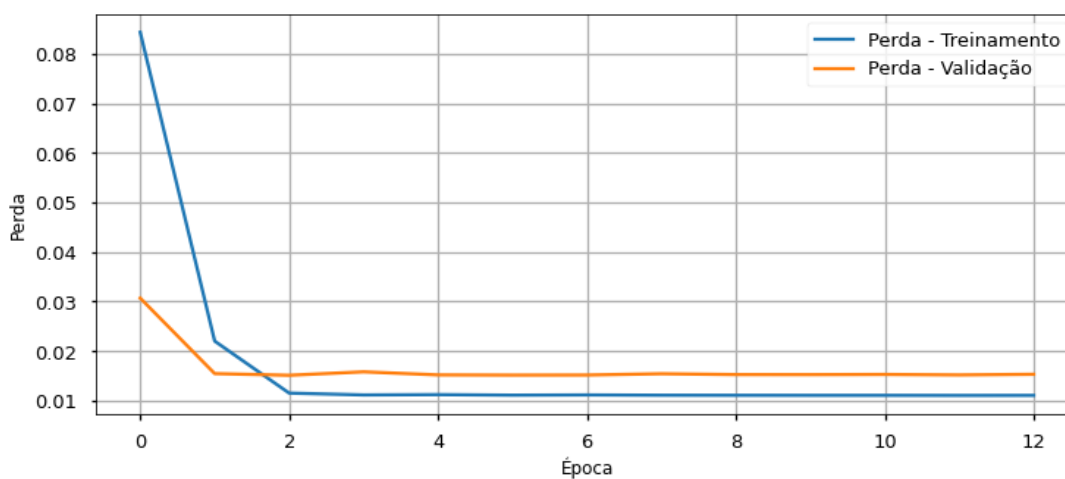
Tabela 11 – Hiperparâmetros utilizados no modelo LSTM padrão.

Atributos	Neurônios	Épocas	Tamanho do lote	Ativação	Otimizador
Univariável	64	12	32	Sigmoide	Adam
Multivariável	64	31	32	Sigmoide	Adam

Fonte: Autor, 2022

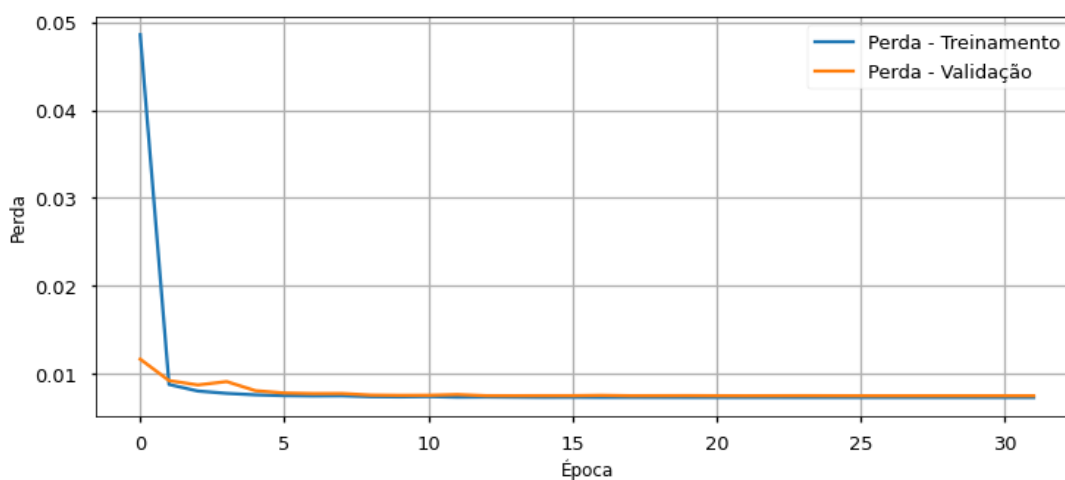
Conforme se pode observar na Figura 32, a LSTM padrão estabilizou com apenas 12 épocas para o treino com uma única variável e em 31 épocas para múltiplos atributos na entrada (Figura 33).

Figura 32 – Perdas (MSE) para LSTM com uma única variável de entrada.



Fonte: (Autor, 2022)

Figura 33 – Perdas (MSE) para LSTM com múltiplas variáveis de entrada.



Fonte: (Autor, 2022)

5.2.3.6 MODELO LSTM COM MÉTODO WINDOW

Assim como foi feito na rede MLP *Window*, na LSTM com método de janela, a cada iteração, para tentar prever o atributo alvo, o modelo irá olhar para 48 informações passadas, o equivalente a 1 hora de registro dos dados na planta fotovoltaica. Os hiperparâmetros escolhidos para o treinamento da rede podem ser vistos na Tabela 12.

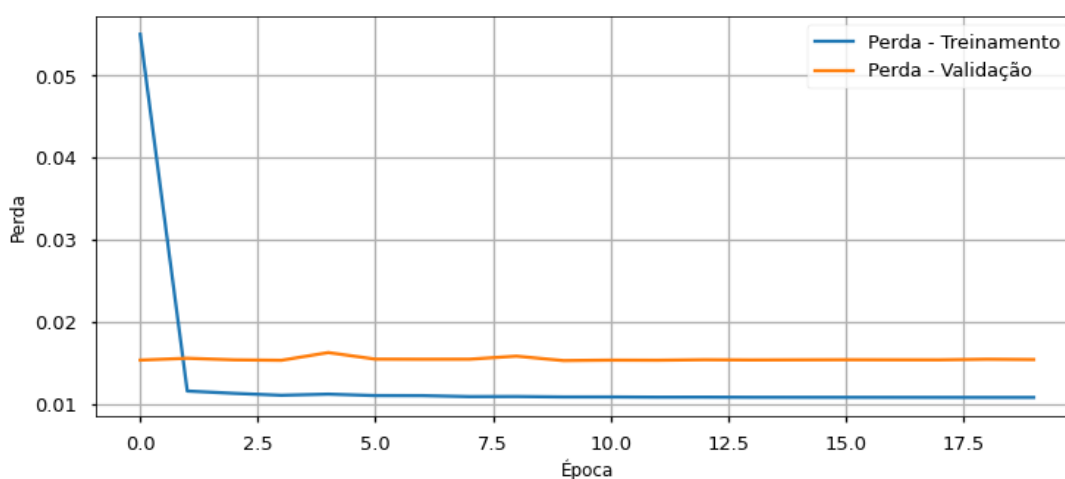
Tabela 12 – Hiperparâmetros utilizados no modelo LSTM com método *Window*.

Atributos	Neurônios	Épocas	Tamanho do lote	Ativação	Otimizador
Univariável	64	19	32	Sigmoide	Adam
Multivariável	64	36	32	Sigmoide	Adam

Fonte: Autor, 2022

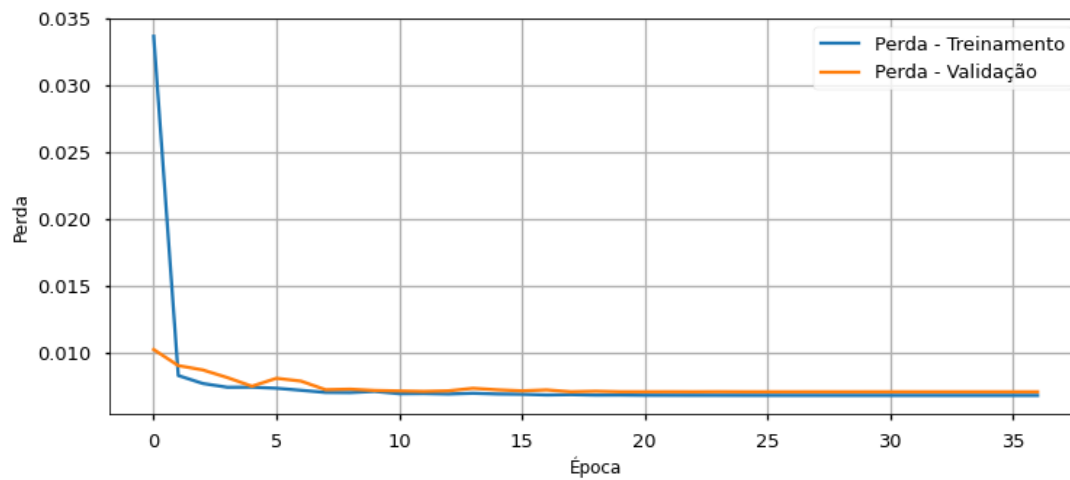
A Figura 34 mostra o histórico de perdas durante toda a etapa de treinamento, percebe-se que ao atingir 19 épocas as perdas se estabilizam, por este motivo, 19 épocas são suficientes para treinar o modelo com uma única variável. Já para o modelo com múltiplos atributos na entrada, conforme mostra Figura 35, a estabilização ocorreu na época 36.

Figura 34 – Perdas (MSE) para LSTM com método *Window* e uma única variável de entrada.



Fonte: (Autor, 2022)

Figura 35 – Perdas (MSE) para LSTM com método *Window* e múltiplas variáveis de entrada.



Fonte: (Autor, 2022)

6 ANÁLISE E DISCUSSÃO DOS RESULTADOS

O resultado obtido com o treinamento de cada um dos modelos de AM discutidos na Seção 5.2 serão apresentados nas seções a seguir.

6.1 PREVISÃO COM MODELOS LINEARES

Os algoritmos de AM considerados nesta etapa utilizam os seguintes modelos lineares:

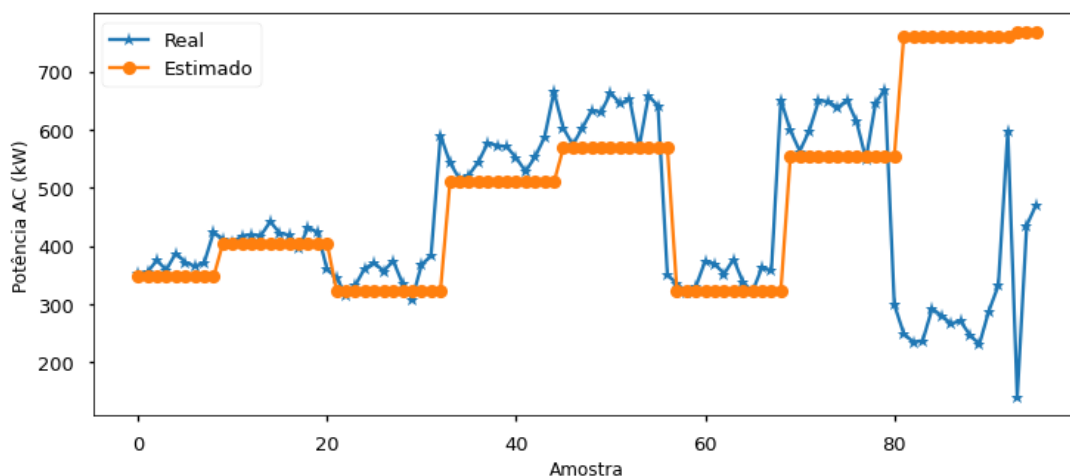
- Modelo de regressão linear unidimensional, ou seja, usando um único atributo como entrada (conforme o Caso 3 da Tabela 8);
- Modelo de regressão linear n-dimensional, ou seja, usando múltiplos atributos como entrada (conforme o Caso 1 da Tabela 8);
- Modelo de regressão polinomial unidimensional;
- Modelo de regressão polinomial n-dimensional.

6.1.1 PREVISÃO REGRESSÃO LINEAR

A Figura 36 exibe o resultado obtido para a previsão dos dados de teste para 96 previsões, que é o equivalente a um período de 2 horas de geração de potência na planta. Fica evidente a baixa resolução do modelo de regressão linear, visto que o modelo gera uma equação da reta e tenta aproximar com segmentos de reta os resultados reais.

Percebe-se que ocorreu um efeito de atraso nos dados estimados, este é um comportamento característico dos algoritmos de AM baseados em modelos de regressão linear, quando o sinal real sofre uma alteração, na previsão seguinte o modelo tenta compensar esta mudança, tentando acompanhar a curva real. A partir da previsão de número 80, percebeu-se que o modelo realizou uma previsão grosseira, que pode ser devido à variação abrupta em um curto espaço de tempo.

Figura 36 – Desempenho no teste para a Regressão Linear com uma única variável de entrada.



Fonte: (Autor, 2022)

A equação da reta obtida pelo modelo e que representa a curva estimada na Figura 36, é dada pela Equação (23).

$$\hat{y} = 0,02329712 + 0,95676469x \tag{23}$$

A Tabela 13 apresenta o erro quadrático médio obtido com os dados normalizados, e o RMSE obtido com os dados não normalizados. Com o RMSE se pôde constatar que, para novos dados, o desvio padrão da potência AC para o modelo de regressão linear é de 121,97 kW dos dados previstos em relação a média dos dados reais (considerando todo o conjunto de teste que totaliza uma média de 419,78 kW de potência AC gerada, o erro RMSE de 121,97 kW corresponde a uma dispersão de 29,05% em relação ao conjunto total de teste).

Tabela 13 – MSE e RMSE obtidos com a Regressão Linear para a previsão dos dados de treino, teste e amostra completa.

Erro	Treino	Teste	Completo
MSE	0,01170	0,00783	0,01090
RMSE	149,16	121,97	143,97

Fonte: Autor, 2022

A Figura 37 exibe um panorama da previsão realizada para todos os 34 dias utilizando o modelo de regressão linear unidimensional considerando o somatório da potência gerada durante cada um dos 34 dias investigados. Em verde está representada a curva real, o comportamento em azul representa a potência prevista utilizando os dados de treino e em laranja a curva representa a potência prevista para os dados novos, referentes ao conjunto de teste.

Figura 37 – Previsão para os 34 dias com modelo de Regressão Linear com uma única variável de entrada.

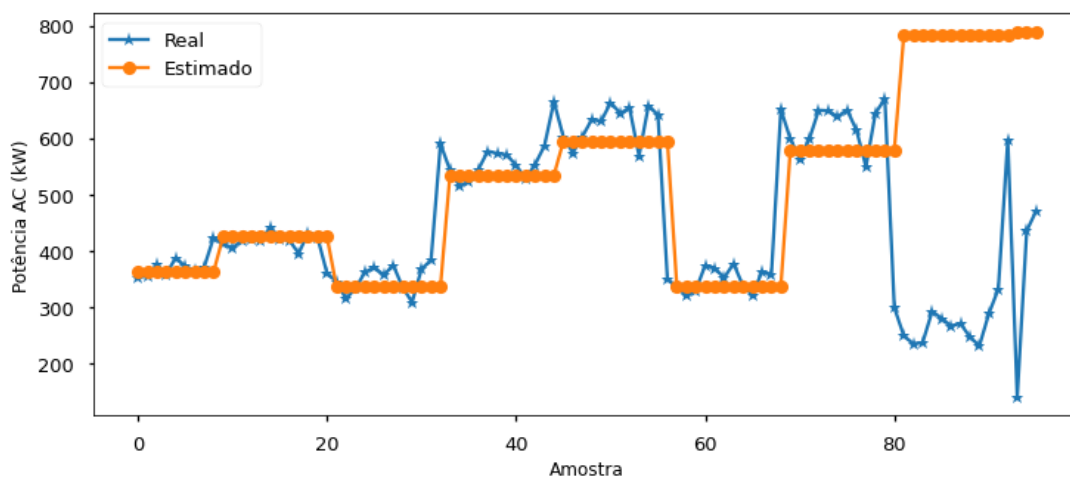


Fonte: (Autor, 2022)

6.1.2 PREVISÃO REGRESSÃO POLINOMIAL

O desempenho para um intervalo de duas horas da amostra de teste pode ser observado na Figura 38. Vale notar que mesmo com sua característica quadrática, a função polinomial ainda apresenta um comportamento constante na saída estimada, embora que levemente mais satisfatório do que o apresentado pelo modelo de regressão linear unidimensional. Isto se dá pelo fato de se ter somente um único atributo de entrada (irradiação) que apresenta pouca variabilidade. Esta característica poderá ser observada nos demais modelos para os casos que se utilizaram de um único atributo na entrada.

Figura 38 – Desempenho no teste para a regressão polinomial com uma única variável de entrada.



Fonte: (Autor, 2022)

A Tabela 14 apresenta o erro quadrático médio obtido com os dados normalizados, e o RMSE obtido com os dados não normalizados. Apresentando uma previsão de geração com desvio padrão de 119,34 kW em relação aos dados reais de geração.

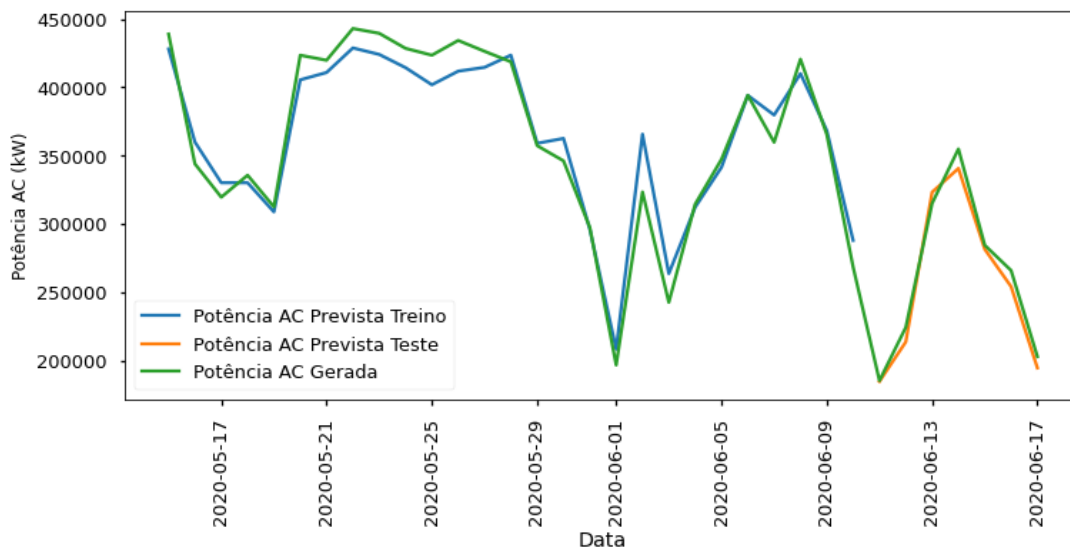
Tabela 14 – MSE e RMSE obtidos com a Regressão Polinomial para a previsão dos dados de treino, teste e amostra completa.

Erro	Treino	Teste	Completo
MSE	0,01149	0,00749	0,01066
RMSE	147,77	119,34	142,37

Fonte: Autor, 2022

A Figura 39 mostra o somatório de potência gerada para cada um dos 34 dias, com a previsão realizada utilizando o modelo de regressão polinomial com uma única variável de entrada.

Figura 39 – Previsão para os 34 dias com modelo de regressão polinomial com uma única variável de entrada.

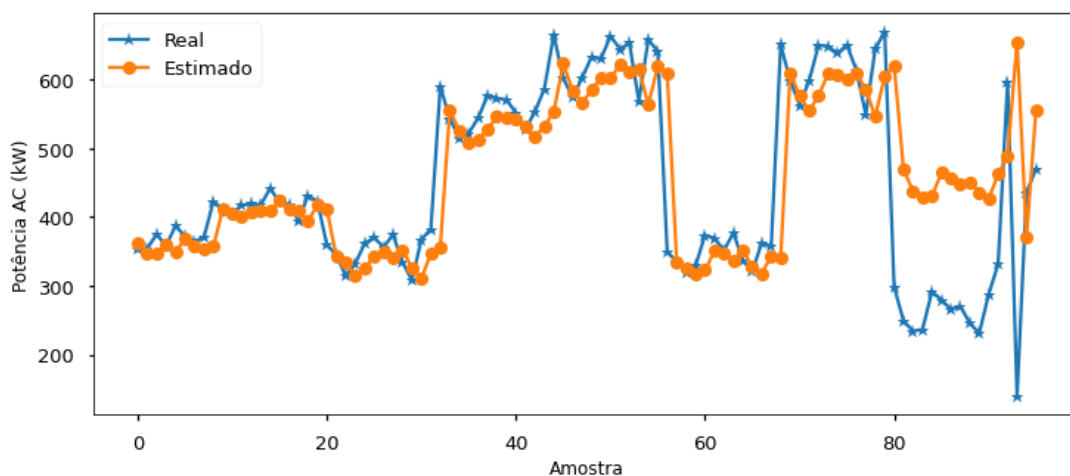


Fonte: (Autor, 2022)

6.1.3 PREVISÃO REGRESSÃO LINEAR COM MULTIVARIÁVEIS

Na Figura 40 consta o resultado obtido para 96 previsões do conjunto de teste com o modelo de regressão linear multivariável. Aqui, como se pode confirmar com a Equação (24) gerada pelo modelo, cada atributo de entrada possui um coeficiente angular e, por este motivo, a previsão de saída deixa de ser uma simples reta e passa a ser um hiperplano, aproximando-se melhor da curva obtida com os dados reais.

Figura 40 – Desempenho no teste para a Regressão Linear com múltiplas variáveis de entrada.



Fonte: (Autor, 2022)

$$\bar{y} = 0,00373198 + 0,026886x_1 - 0,01202619x_2 + 0,35961107x_3 + 0,00558396x_4 - 0,00944911x_5 + 0,61949357x_6 \quad (24)$$

A Tabela 15 reúne o erro quadrático médio obtido com os dados normalizados, e o RMSE obtido com os dados não normalizados. Obtendo um desvio padrão total dos testes na geração de potência com novos dados de 87,86 kW em relação a média dos dados reais.

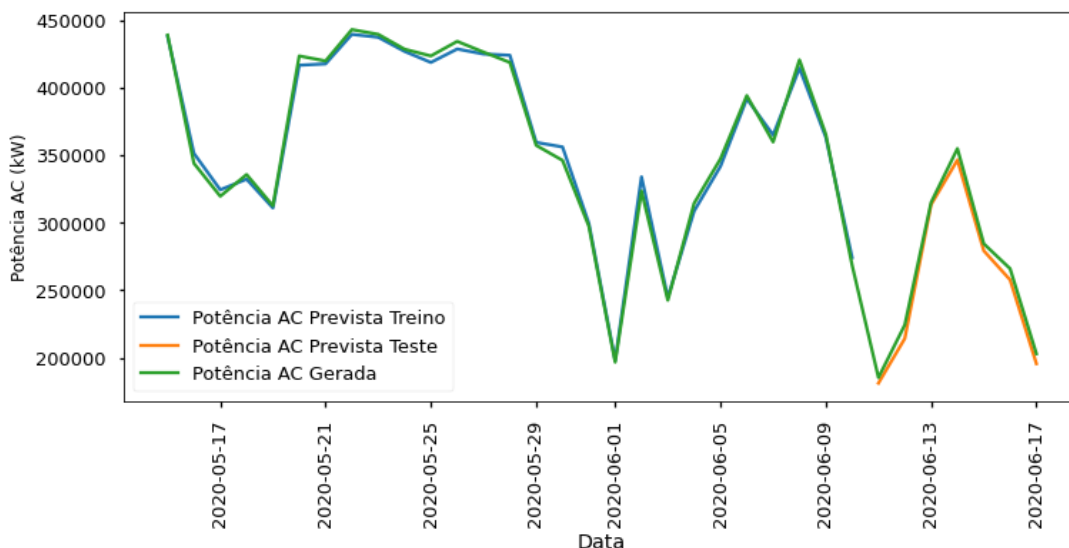
Tabela 15 – MSE e RMSE obtidos com a Regressão Linear com múltiplas variáveis para a previsão dos dados de treino, teste e amostra completa.

Erro	Treino	Teste	Completo
MSE	0,00733	0,00406	0,00666
RMSE	118,06	87,86	112,49

Fonte: Autor, 2022

Segue na Figura 41 um panorama da previsão realizada para todos os 34 dias utilizando o modelo de regressão linear com múltiplas variáveis, com o somatório de potência gerada para cada um dos dias.

Figura 41 – Previsão para os 34 dias com modelo de Regressão Linear com múltiplas variáveis de entrada.



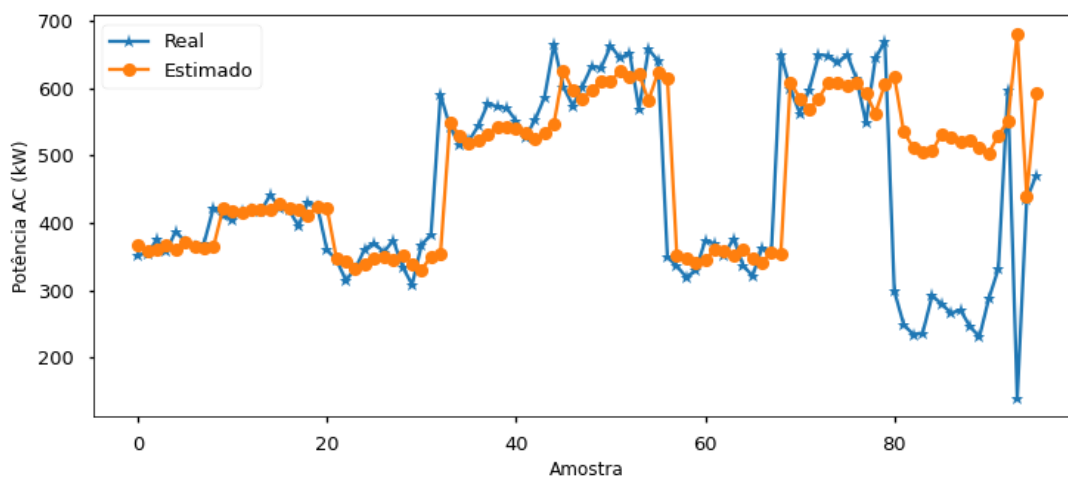
Fonte: (Autor, 2022)

6.1.4 PREVISÃO REGRESSÃO POLINOMIAL COM MULTIVARIÁVEIS

Assim como nos casos anteriores, na Figura 42 se tem o desempenho de 96 previsões para a amostra de teste, agora com o modelo de regressão polinomial com

múltiplas variáveis. Neste caso, cada um dos atributos de entrada são elevados ao quadrado, possibilitando uma característica de parábola para as curvas estimadas, resultando em uma aproximação com maior fidelidade em relação aos dados reais.

Figura 42 – Desempenho no teste para a regressão polinomial com múltiplas variáveis de entrada.



Fonte: (Autor, 2022)

Na regressão polinomial, como discutido na Seção 4.5, considerando um polinômio de grau 2 ($p = 2$) para a transformação dos dados de entrada, é realizado o produto entre cada atributo de entrada. Assim, os 6 atributos de entrada são transformados em 12 atributos e fornecidos para o treinamento do modelo.

A Tabela 16 apresenta o erro quadrático médio obtido com os dados normalizados, e o RMSE obtido com os dados não normalizados. Sendo assim, para novos dados, o modelo de regressão polinomial apresenta um desvio padrão de 87,58 kW na previsão em relação a média dos dados reais.

Tabela 16 – Resultados obtidos com a Regressão Polinomial com múltiplas variáveis para a previsão dos dados de treino, teste e amostra completa.

Erro	Treino	Teste	Completo
MSE	0,00689	0,00403	0,00630
RMSE	114,41	87,58	109,41

Fonte: Autor, 2022

A Figura 43 mostra o somatório de potência gerada para cada um dos 34 dias, a previsão realizada com o modelo de regressão polinomial com múltiplas variáveis.

Figura 43 – Previsão para os 34 dias com modelo de regressão polinomial com múltiplas variáveis de entrada.



Fonte: (Autor, 2022)

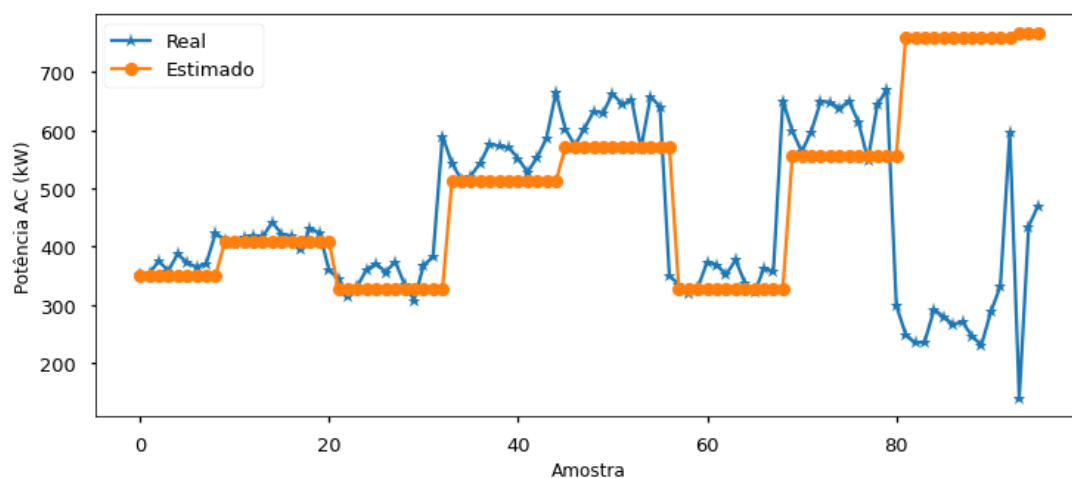
6.2 PREVISÃO COM REDES NEURAIS

A seguir serão apresentados os resultados obtidos com as duas redes neurais treinadas, a rede neural MLP e LSTM. São reunidos os resultados para os treinamentos com os dois casos estudados, com uma única variável e com múltiplas variáveis de entrada, para cada um dos casos foram aplicados o método *Vanilla* e *Window*, resultando em um total de quatro redes neurais modeladas.

6.2.1 PREVISÃO MLP

O comportamento previsto para duas horas do conjunto de teste apresentado pela primeira rede neural utilizando somente a irradiação como entrada pode ser observado na Figura 44. Assim como foi visto nos resultados da regressão polinomial, aqui o comportamento da previsão tornou-se aparentemente linear, dada a baixa variabilidade do atributo de irradiação.

Figura 44 – Desempenho no teste para a MLP com uma única variável de entrada.



Fonte: (Autor, 2022)

Na Tabela 17 estão mostrados os valores para o erro quadrático médio obtido com os dados normalizados, e o RMSE obtido com os dados não normalizados. Sendo assim, se tem que para os novos dados de entrada, obteve-se um desvio padrão de 121,65 kW dos resultados previstos para a média dos dados reais.

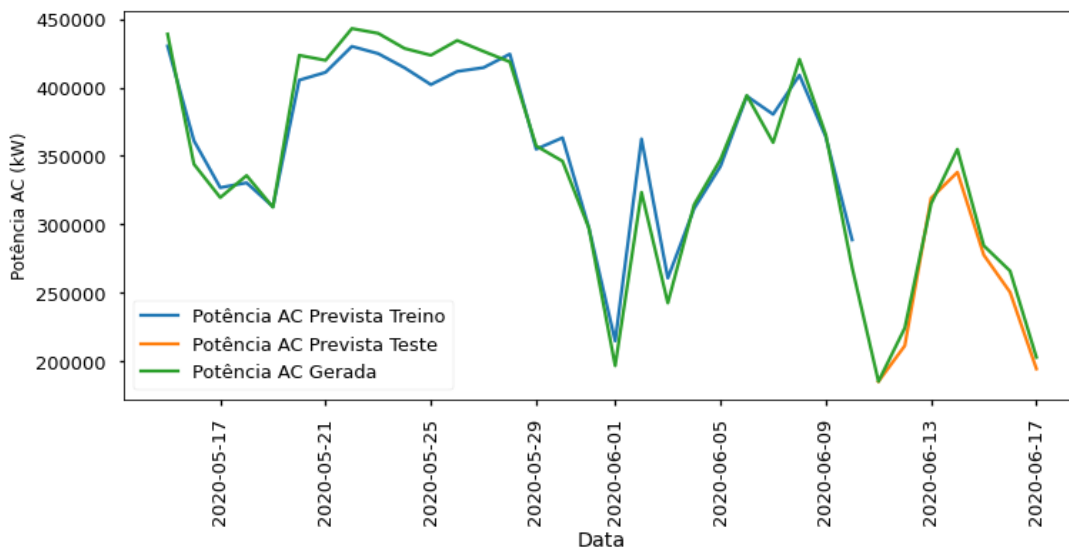
Tabela 17 – MSE e RMSE obtidos com a rede neural MLP com uma única variável para a previsão dos dados de treino, teste e amostra completa.

Erro	Treino	Teste	Completo
MSE	0,01169	0,00778	0,01088
RMSE	149,06	121,65	143,83

Fonte: Autor, 2022

A Figura 45 apresenta o somatório de potência gerada para cada um dos 34 dias por todos os inversores considerados, comparando com a previsão resultante do modelo de rede neural MLP utilizando um único atributo de irradiação na entrada.

Figura 45 – Previsão para os 34 dias com modelo de rede neural MLP com uma única variável de entrada.

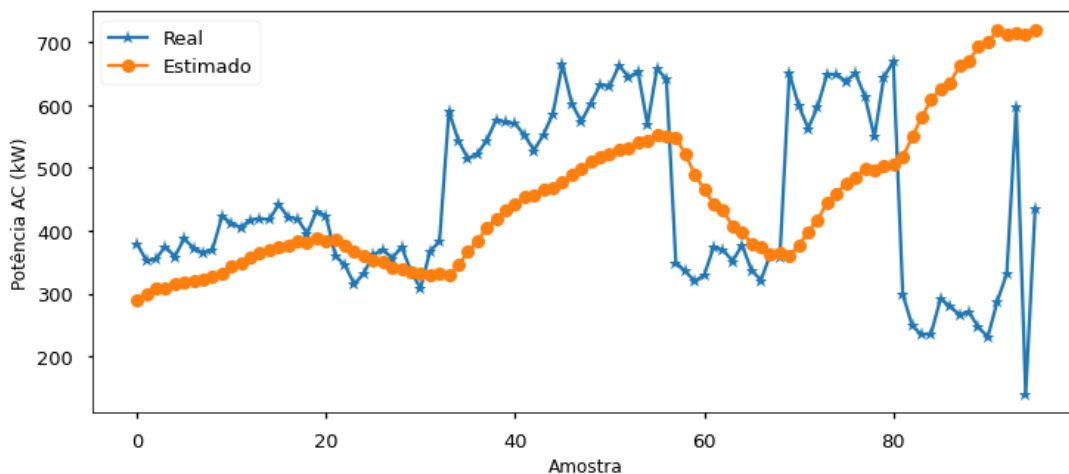


Fonte: (Autor, 2022)

6.2.2 PREVISÃO MLP MÉTODO WINDOW

Aplicando o método de janela na rede neural MLP, olhando para 48 passos anteriores, obteve-se o desempenho mostrado pela Figura 46 para a um intervalo de duas horas da amostra de teste.

Figura 46 – Desempenho no teste para a MLP com método *Window* e uma única variável de entrada.



Fonte: (Autor, 2022)

Na Tabela 18 estão mostrados os resultados para o erro quadrático médio obtido com os dados normalizados, e o RMSE obtido com os dados não normalizados.

Sendo assim, se tem que para os novos dados de entrada, obteve-se um desvio padrão de 136,82 kW dos resultados previstos para os reais.

Tabela 18 – MSE e RMSE obtidos com a rede neural MLP com método *Window* e uma única variável para a previsão dos dados de treino, teste e amostra completa.

	Erro	Treino	Teste	Completo
MSE	0,01223	0,00985	0,01171	
RMSE	152,46	136,82	149,22	

Fonte: Autor, 2022

A Figura 47 apresenta o somatório de potência gerada para cada um dos 34 dias por todos os inversores considerados, comparando com a previsão resultante do modelo de rede neural MLP com um único atributo na entrada.

Figura 47 – Previsão para os 34 dias com modelo de rede neural MLP com método *Window* e uma única variável de entrada.

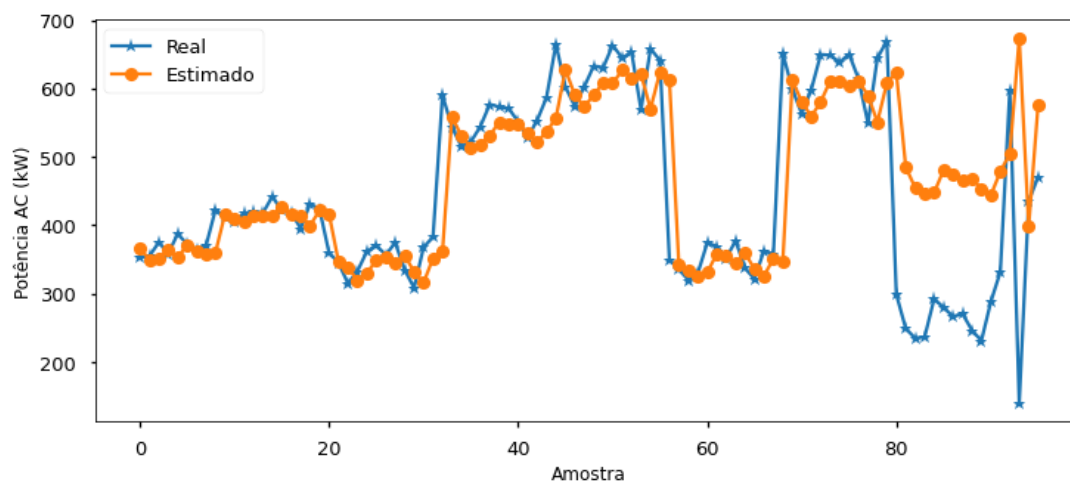


Fonte: (Autor, 2022)

6.2.3 PREVISÃO MLP COM MULTIVARIÁVEIS

O desempenho apresentado pela MLP padrão com múltiplas variáveis na amostra de teste pode ser observado na Figura 48.

Figura 48 – Desempenho no teste para a MLP com múltiplas variáveis de entrada.



Fonte: (Autor, 2022)

Na Tabela 19 percebe-se que a potência para a amostra com os novos dados variam em uma média de 88,06 kW dos resultados previstos para os reais.

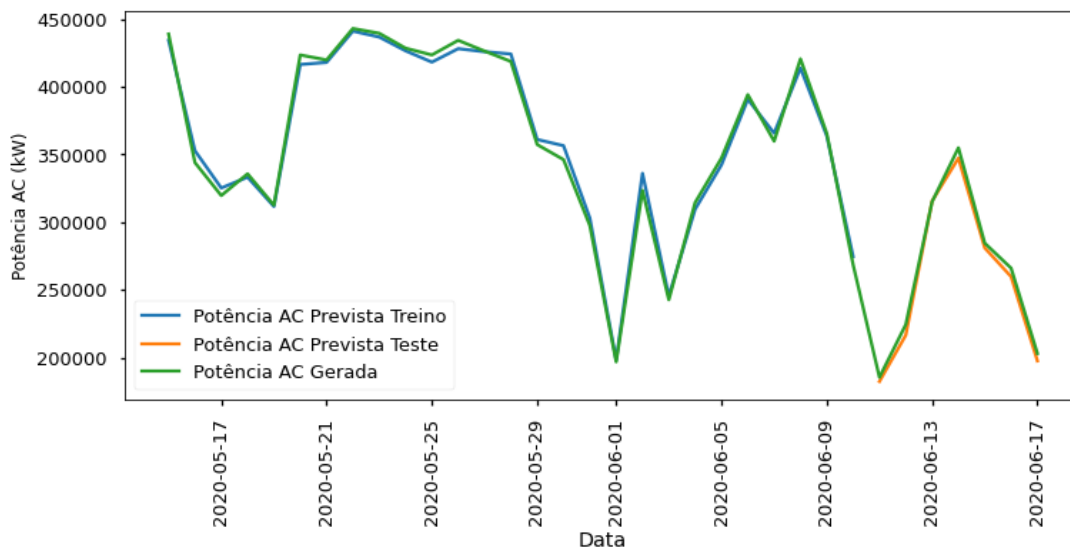
Tabela 19 – MSE e RMSE obtidos com a rede neural MLP para múltiplas variáveis para a previsão dos dados de treino, teste e amostra completa.

Erro	Treino	Teste	Completo
MSE	0,00734	0,00408	0,00666
RMSE	118,10	88,06	112,56

Fonte: Autor, 2022

A previsão para o total de potência gerada em cada um dos trinta e quatro dias para o modelo em questão pode ser observada na Figura 49.

Figura 49 – Previsão para os 34 dias com modelo de rede neural MLP com múltiplas variáveis de entrada.

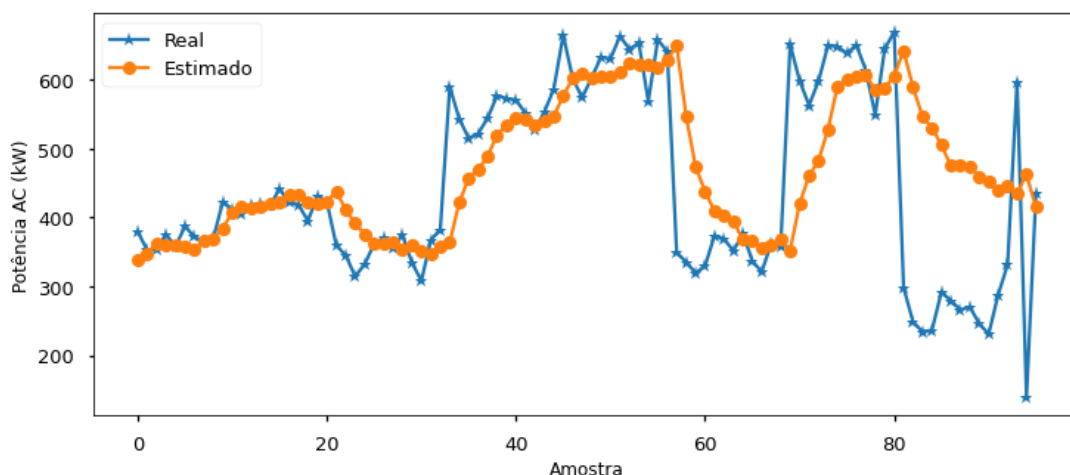


Fonte: (Autor, 2022)

6.2.4 PREVISÃO MLP MÉTODO WINDOW COM MULTIVARIÁVEIS

Para a rede MLP com método de janela e múltiplos atributos na entrada, o desempenho apresentado para um intervalo da amostra de teste é o apresentado na Figura 50.

Figura 50 – Desempenho no teste para a MLP com método *Window* e múltiplas variáveis de entrada.



Fonte: (Autor, 2022)

Os resultados de MSE para o modelo MLP com método de janela estão reunidos na Tabela 20. Observa-se um desvio padrão de 92,26 kW dos resultados previstos para os reais.

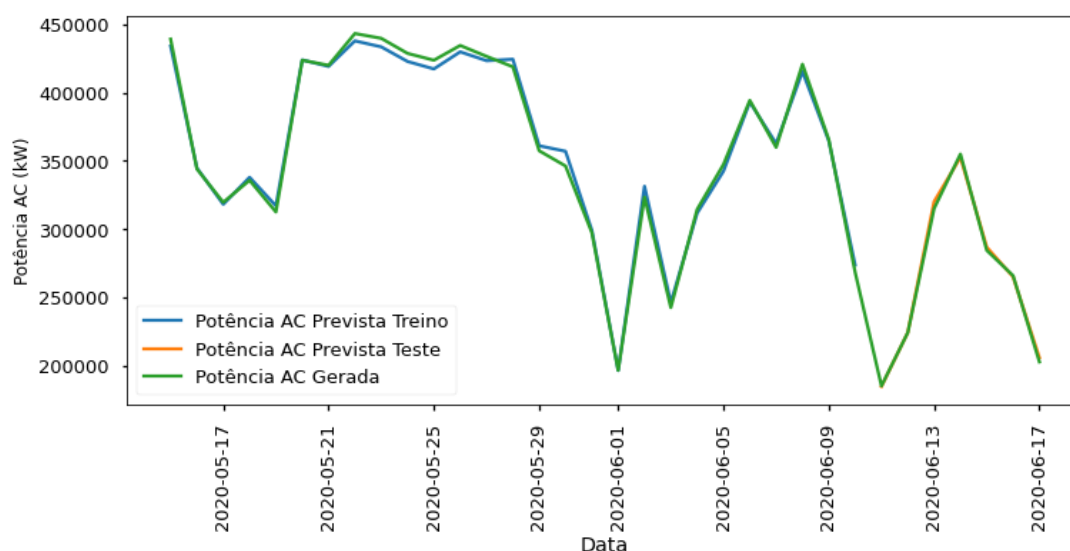
Tabela 20 – MSE e RMSE obtidos com a rede neural MLP com método *Window* para múltiplas variáveis para a previsão dos dados de treino, teste e amostra completa.

Erro	Treino	Teste	Completo
MSE	0,00712	0,00448	0,00657
RMSE	116,37	92,26	111,73

Fonte: Autor, 2022

A previsão da geração de potência AC diária durante todo período observado, aparece na Figura 51.

Figura 51 – Previsão para os 34 dias com modelo de rede neural MLP com método *Window* e múltiplas variáveis de entrada.

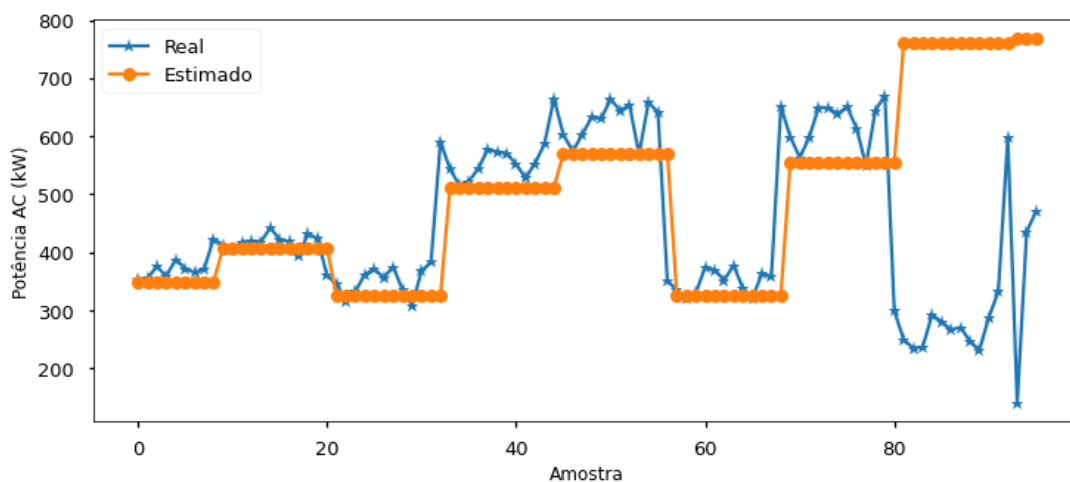


Fonte: (Autor, 2022)

6.2.5 PREVISÃO LSTM

O modelo LSTM padrão treinado somente com o atributo de irradiação na entrada teve o desempenho mostrado pela Figura 52, para um intervalo de duas horas da amostra de teste.

Figura 52 – Desempenho no teste para a LSTM com uma única variável de entrada.



Fonte: (Autor, 2022)

Os valores de MSE para o modelo em questão estão na Tabela 21.

Tabela 21 – MSE e RMSE obtidos com a rede neural LSTM com uma única variável para a previsão dos dados de treino, teste e amostra completa.

Erro	Treino	Teste	Completo
MSE	0,01169	0,00780	0,01089
RMSE	149,09	121,79	143,88

Fonte: Autor, 2022

A previsão para os 34 dias do modelo pode ser comparada com os valores reais na Figura 53.

Figura 53 – Previsão para os 34 dias com modelo de rede neural LSTM com uma única variável de entrada.

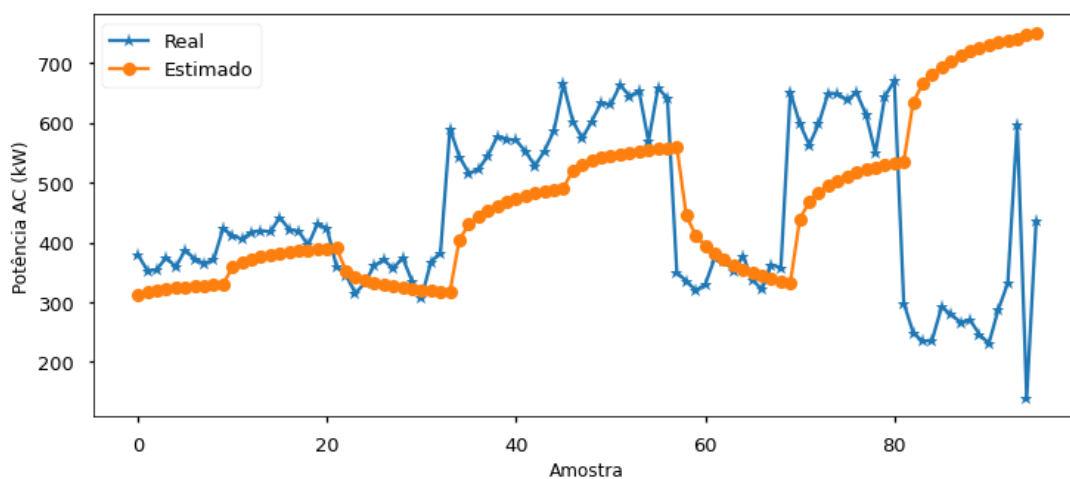


Fonte: (Autor, 2022)

6.2.6 PREVISÃO LSTM MÉTODO WINDOW

Para finalizar os treinamentos com um único atributo na entrada, se tem o desempenho para a amostra de teste com a LSTM com método de janela na Figura 54.

Figura 54 – Desempenho no teste para a LSTM com método *Window* e uma única variável de entrada.



Fonte: (Autor, 2022)

Os erros apresentados para este modelo podem ser observados na Tabela 22.

Tabela 22 – MSE e RMSE obtidos com a rede neural LSTM com método *Window* e uma única variável para a previsão dos dados de treino, teste e amostra completa.

	Erro	Treino	Teste	Completo
MSE	0,01153	0,00846	0,01087	
RMSE	148,03	126,81	143,77	

Fonte: Autor, 2022

O desempenho apresentado durante todos os trinta e quatro dias pelo treinamento da LSTM método de janela, da soma de geração de potência AC diária, está na Figura 55.

Figura 55 – Previsão para os 34 dias com modelo de rede neural LSTM com método *Window* e uma única variável de entrada.

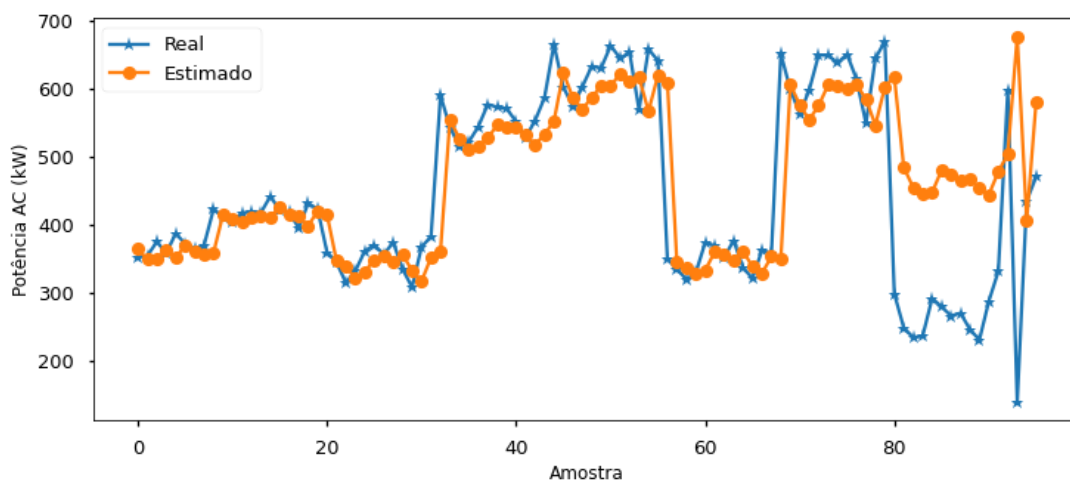


Fonte: (Autor, 2022)

6.2.7 PREVISÃO LSTM COM MULTIVARIÁVEIS

A LSTM padrão com múltiplos atributos de entrada apresentou o desempenho para um intervalo da amostra de teste conforme Figura 58.

Figura 56 – Desempenho no teste para a LSTM com múltiplas variáveis de entrada.



Fonte: (Autor, 2022)

A Tabela 24 reúne o erro quadrático médio obtido com os dados normalizados, e o RMSE obtido com os dados não normalizados, o que resulta em um desvio padrão de 88,70 kW dos dados previstos em relação aos dados reais.

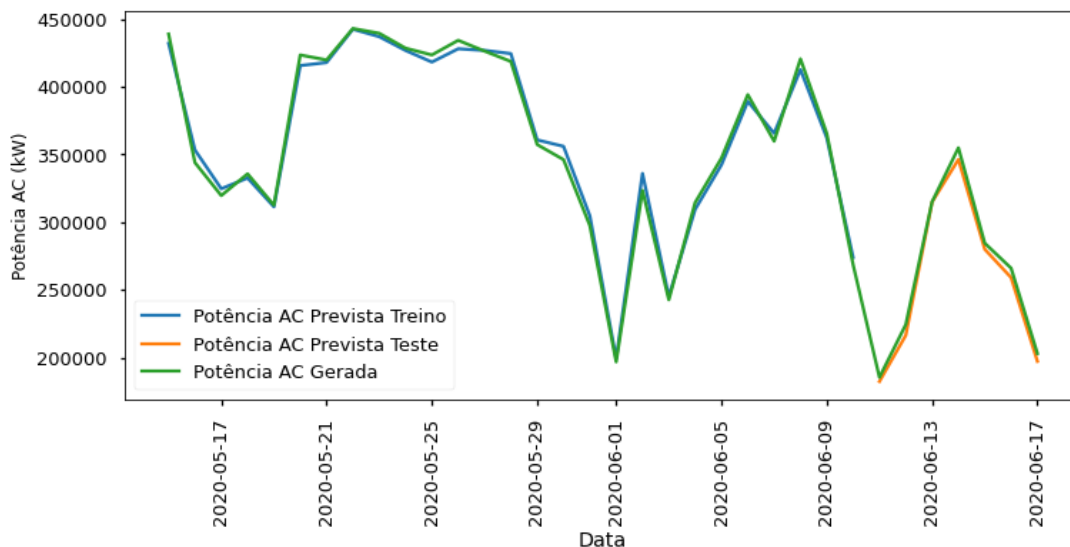
Tabela 23 – MSE e RMSE obtidos com a rede neural LSTM para múltiplas variáveis para a previsão dos dados de treino, teste e amostra completa.

Erro	Treino	Teste	Completo
MSE	0,00735	0,00414	0,00669
RMSE	118,23	88,70	112,77

Fonte: Autor, 2022

A previsão de potência elétrica utilizando o modelo de rede neural LSTM com múltiplas variáveis de entrada pode ser vista na Figura 57.

Figura 57 – Previsão para os 34 dias com modelo de rede neural LSTM com múltiplas variáveis de entrada.

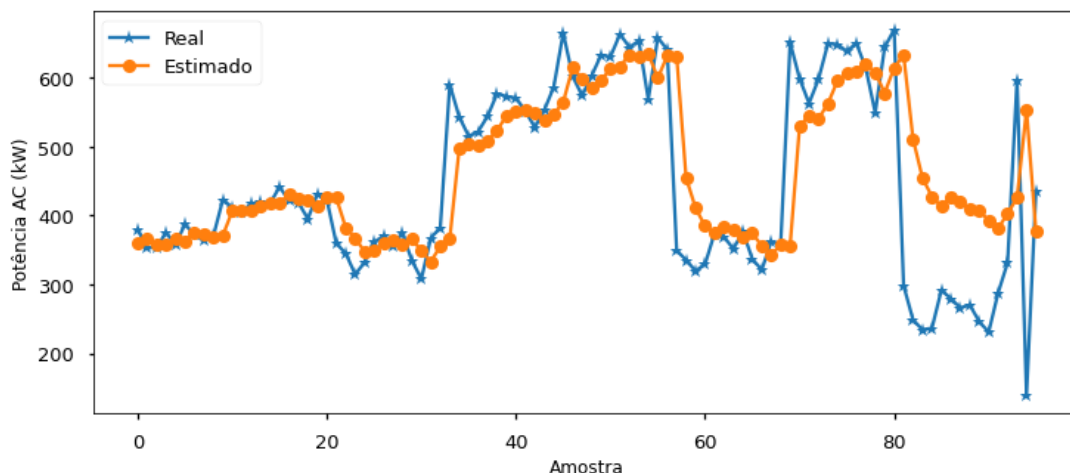


Fonte: (Autor, 2022)

6.2.8 PREVISÃO LSTM MÉTODO WINDOW COM MULTIVARIÁVEIS

Para o modelo LSTM com método de janela e múltiplas entradas, o desempenho apresentado para um intervalo de duas horas da amostra de teste pode ser observado na Figura 58.

Figura 58 – Desempenho no teste para a LSTM com método *Window* e múltiplas variáveis de entrada.



Fonte: (Autor, 2022)

O MSE obtido com os dados normalizados, e o RMSE obtido com os dados não normalizados para o modelo LSTM com método de janela e múltiplas variáveis de

entrada estão reunidos na Tabela 24. O desvio padrão da potência elétrica prevista em relação aos dados reais é de 83,98 kW.

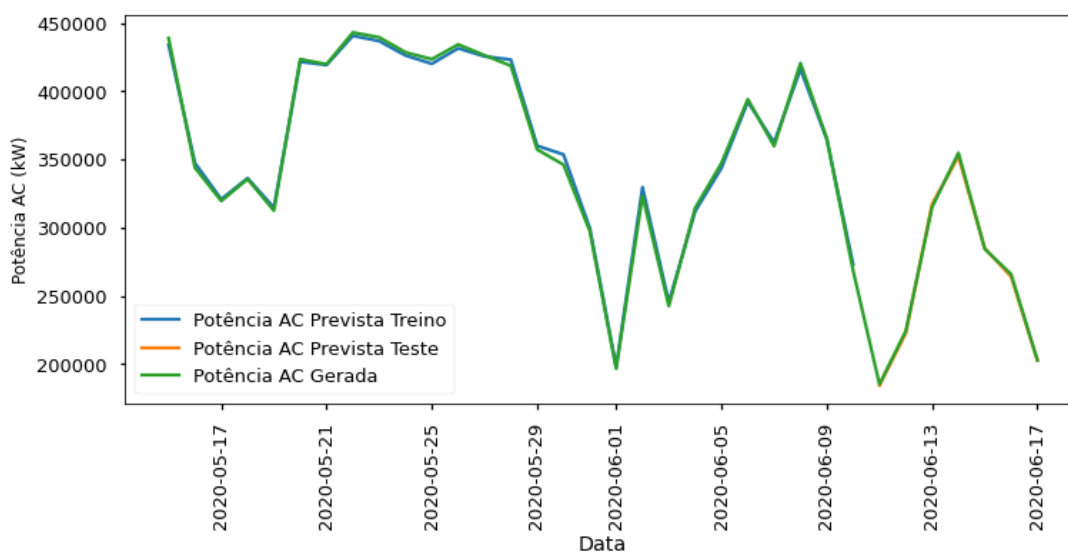
Tabela 24 – MSE e RMSE obtidos com a rede neural LSTM com método *Window* para múltiplas variáveis para a previsão dos dados de treino, teste e amostra completa.

	Erro	Treino	Teste	Completo
MSE	0,00682	0,00371	0,00617	
RMSE	113,90	83,98	108,32	

Fonte: Autor, 2022

Por fim, a previsão de potência elétrica utilizando o modelo de rede neural LSTM com método de janela e múltiplos atributos de entrada pode ser vista na Figura 59.

Figura 59 – Previsão para os 34 dias com modelo de rede neural LSTM com método *Window* e múltiplas variáveis de entrada.



Fonte: (Autor, 2022)

6.3 COMPARAÇÃO DOS MODELOS APLICADOS

Na Tabela 25 estão reunidos os resultados de erro quadrático médio obtidos com o treinamento de todos os modelos para o Caso 1, com seis atributos na entrada do treinamento. Como esperado, a regressão linear apresentou o resultado menos satisfatório, com o maior erro quadrático médio para a amostra de treino e amostra completa, no entanto, apresentou um erro para a amostra de teste mais satisfatório do que aquele apresentado pela rede neural MLP, que inclusive, registrou um desempenho pior do que o apresentado pela regressão polinomial.

Considerando a baixa quantidade de dados disponível para o treinamento dos algoritmos de AM baseados em redes neurais e observando os valores satisfatórios dos erros obtidos pelos modelos de regressão linear, infere-se que os modelos de redes neurais (que calculam os pesos dos seus correspondentes coeficientes iterativamente através do método do Gradiente Descendente) não alcançaram todo o potencial oferecido por sua arquitetura. Visto que, para múltiplos atributos de entrada, o modelo de regressão linear obteve um desempenho melhor do que aquele obtido pela rede neural MLP sobre o conjunto de teste. A rede neural MLP, apesar de não ter apresentado o pior resultado no conjunto completo, apresentou o pior MSE no conjunto de teste com o método de janela.

O melhor desempenho ficou com a rede neural LSTM utilizando o método *window*, que de forma expressiva obteve o menor erro quadrático médio para os dados do conjunto de teste. Isso pode ter sido resultado da possibilidade de transformação de atributos a partir da combinação dos atributos correntes e também dos atributos prévios presente na janela de observação (método *window*). Desse modo, na saída de cada camada da rede neural uma nova transformação de atributos é realizada, permitindo a criação de atributos mais significativos para a tarefa de estimação da potência elétrica gerada.

Tabela 25 – Erro quadrático médio normalizado obtido para todos os modelos treinados com múltiplas variáveis de entrada.

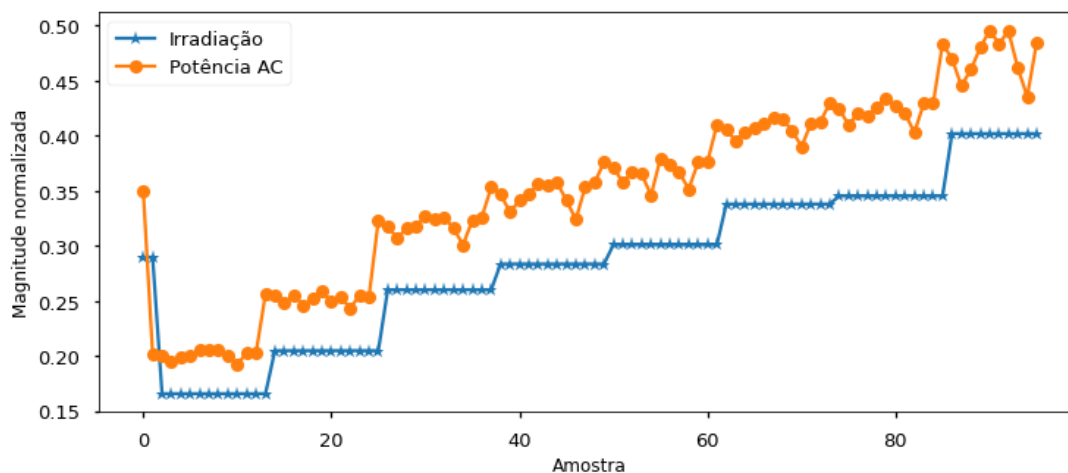
Modelo	Treino	Teste	Completo
Regressão Linear	0,00733	0,00406	0,00666
Regressão Polinomial	0,00689	0,00403	0,00630
MLP	0,00734	0,00408	0,00666
MLP Window	0,00712	0,00448	0,00657
LSTM	0,00735	0,00414	0,00669
LSTM Window	0,00682	0,00371	0,00617

Fonte: Autor, 2022

Para os modelos treinados com um único atributo de entrada, pôde-se perceber que todos apresentaram um comportamento constante durante um certo intervalo de tempo, na previsão realizada, com exceção das redes neurais utilizando o método de janela, visto que nesses métodos as entradas possuíam mais atributos para serem tratados e transformados. Os valores constantes apresentados por esses casos, se dá pelo fato da irradiação ser um atributo com pouca variabilidade, a Figura 60 exemplifica esta situação, relacionando a curva da entrada (irradiação) com a curva da saída (potência AC), percebe-se que a potência AC possui uma variabilidade maior do que a entrada, portanto, visto que para um mesmo valor de entrada se tem diferentes valores de saída, o modelo faz uma média destas saídas, resultando em uma estimação

constante durante certos intervalos.

Figura 60 – Comparativo entre o atributo de entrada e de saída do modelo com uma única variável de entrada.



Fonte: (Autor, 2022)

Para um único atributo de entrada dos modelos (Tabela 26), o de regressão polinomial apresentou o melhor desempenho, evidenciando que para a amostra de dados trabalhada, o uso de modelos de regressão simples é suficiente para um resultado satisfatório.

Tabela 26 – Erro quadrático médio normalizado obtido para todos os modelos treinados com uma única variável de entrada.

Modelo	Treino	Teste	Completo
Regressão Linear	0,01170	0,00783	0,01090
Regressão Polinomial	0,01149	0,00749	0,01066
MLP	0,01169	0,00778	0,01088
MLP Window	0,01223	0,00985	0,01171
LSTM	0,01169	0,00780	0,01089
LSTM Window	0,01153	0,00846	0,01087

Fonte: Autor, 2022

Como já citado no decorrer do trabalho, para desenvolver os modelos de regressão linear foi utilizada a biblioteca *ScikitLearn*. Essa biblioteca utiliza o método dos mínimos quadrados para realizar o treinamento do modelo, ou seja, o modelo tenta encontrar a reta que apresenta o menor erro quadrático médio em relação aos dados reais, logo, esta é uma função de característica analítica e que pode apresentar um ótimo desempenho para um baixo volume de dados, desde que estes dados apresentem relevância para o treinamento. Além disso, o desempenho satisfatório dos modelos de regressão no presente trabalho evidenciam um tratamento adequado dos dados, realizado na etapa de AED.

A rede neural por sua vez, não utiliza o método dos mínimos quadrados para realizar seu treinamento, aqui é utilizado o método do gradiente descendente através do algoritmo *backpropagation* conforme explicado na Subsubseção 4.6.3.2. Tal Método é iterativo e corresponde a uma aproximação da solução ótima (encontrada via método dos mínimos quadrados), possibilitando a estagnação do gradiente de erro em um mínimo local distante do verdadeiro mínimo global (da função custo $J(\mathbf{w})$). No entanto, a rede neural, possibilita a criação de mais atributos (via transformações lineares ou não lineares realizada pela função de ativação dos neurônios nas saídas de cada camada da rede) durante o treinamento, tornando mais eficiente a sua previsão principalmente com um grande volume de dados, e uma baixa dependência de atributos relevantes, diferente do modelo analítico da regressão linear. A Tabela 27 mostra o desvio padrão da média dos dados reais obtidos com cada um dos modelos treinados para a amostra de teste, com destaque para os que apresentaram melhor performance em relação aos dois diferentes métodos de atributos, com univariável e multivariável.

Tabela 27 – Raíz do erro quadrático médio obtido para todos os modelos treinados para a amostra de teste.

Modelo	Atributos	Método	RMSE
Regressão Linear	Univariável	Mínimos Quadrados	121,97 kW
Regressão Linear	Multivariável		87,86 kW
Regressão Polinomial	Univariável		119,34 kW
Regressão Polinomial	Multivariável		87,58 kW
MLP	Univariável	Gradiente Descendente	121,65 kW
MLP	Multivariável		88,06 kW
MLP Window	Univariável		136,82 kW
MLP Window	Multivariável		92,26 kW
LSTM	Univariável		121,79 kW
LSTM	Multivariável		88,70 kW
LSTM Window	Univariável		126,81 kW
LSTM Window	Multivariável		83,98 kW

Fonte: Autor, 2022

Todos os modelos de rede neural foram treinados com poucas épocas, visto a estabilização rápida destes, o treinamento foi programado para ser interrompido caso, após 10 épocas, não houvesse diminuição do erro quadrático médio para a amostra de validação. Optar por mais épocas, apesar de poder vir a resultar em uma perda menor, poderá ocasionar em um *overfitting* do modelo.

7 CONCLUSÃO E CONSIDERAÇÕES FINAIS

Neste trabalho, uma estrutura de códigos (*framework*) foi desenvolvida com base em análise exploratória de dados e algoritmos de AM (a saber, regressão linear e redes neurais). O *framework* desenvolvido foi aplicado ao problema de estimação de geração de potência elétrica em uma planta fotovoltaica. Especificamente, o *framework* supracitado foi configurado para realizar previsões de curto prazo, visto que seu treinamento foi realizado com dados coletados em um “curto” intervalo de tempo (trinta e quatro dias). Dessa forma, devido à natureza estocástica dos dados e a influência da sazonalidade na irradiação solar, para realizar uma previsão adequada da geração de potência elétrica de um sistema fotovoltaico para longo prazo, por exemplo: um ano, na região em que o sistema fotovoltaico está localizado, será necessário um conjunto de dados consideravelmente maior em relação ao conjunto disponível neste trabalho. Assim, os atuais modelos de AM criados a partir do referido *framework*, na sua configuração atual, funcionarão de forma satisfatória ao serem “alimentados” com conjuntos de dados referentes ao período aqui investigado (entre maio e junho) e proveniente das plantas solares estudadas.

Neste trabalho, os resultados obtidos foram consistentes, confirmando a análise realizada durante a etapa de AED (Subseção 5.1.7), a qual indicou que o modelo “alimentado” com múltiplos atributos de entrada apresentaria um resultado mais satisfatório do que aquele alimentado com um único atributo. Ao comparar os modelos treinados, percebeu-se uma redução expressiva do erro ao realizar o treinamento com múltiplas variáveis. Ainda assim, os modelos univariáveis apresentaram um desempenho adequado, principalmente visto a alta correlação entre os atributos de entrada e saída. No caso de um único atributo de entrada, a regressão polinomial apresentou um melhor desempenho. Para o caso de múltiplos atributos, a rede LSTM com método de janela obteve a melhor performance.

7.1 TRABALHOS FUTUROS

Visando uma maior robustez do conjunto de dados necessário para o treinamento de algoritmos de AM, sugere-se o desenvolvimento de trabalhos acadêmicos para a realização de coleta abundante de dados referentes a novas plantas fotovoltaicas. Nesse contexto, sugere-se a coleta de dados referente a um período mínimo de 6 meses. Após a obtenção de um banco de dados mais robusto, o *framework* desenvolvido aqui neste trabalho poderá ser utilizado e novas investigações sobre as técnicas de AM de regressão linear, MLP e LSTM podem ser realizadas.

Ressalta-se que, a fim de contornar o problema da baixa quantidade de dados

utilizada no presente trabalho, foi realizada uma pesquisa bibliográfica em busca de trabalhos que utilizam conjuntos de dados maiores do que o aplicado neste trabalho. Foi encontrado o trabalho de Santos (2019), onde a previsão da produção de uma usina solar instalada na Universidade Federal de Goiás (UFG), com dados referente a um período de 6 meses, é realizada. Ao entrar em contato com a Universidade, foi fornecido o acesso ao sistema com a base dos dados utilizados, que possui uma quantidade expressiva de amostras (com valores coletados desde outubro de 2018). No entanto, não se obteve sucesso em capturar a base de dados do sistema devido a não disponibilização dos dados para *download*. O acesso permitido apenas possibilitou a visualização de gráficos, sendo necessária uma coleta manual de todos os dados relacionados com as plantas fotovoltaicas. Dessa forma, tal coleta tornou-se inviável de ser realizada neste trabalho de conclusão de curso.

7.2 CONSIDERAÇÕES FINAIS

Considerando a configuração adequada de seus hiperparâmetros, o *framework* desenvolvido aqui neste trabalho está habilitado à previsão de séries temporais disponíveis em qualquer conjunto de dados. Além disso, a fim de incentivar futuras investigações e novas aplicações de algoritmos de AM, disponibiliza-se o *framework* através do seguinte link: <https://github.com/franciscogazaniga/TCC_Aprendizado_Maquina_Previsao_Energia_Fotovoltaica>.

Referências

- ABDEL-NASSER, M.; MAHMOUD, K. Accurate photovoltaic power forecasting models using deep lstm-rnn. **Neural Computing and Applications**, Springer, v. 31, n. 7, p. 2727–2740, 2019. Citado 3 vezes nas páginas 6, 7 e 8.
- BARRETO, J. M. Introdução às redes neurais artificiais. Florianópolis, 2002. Citado na página 16.
- BASTOS, I. G. P. et al. Previsão de geração fotovoltaica a partir de dados meteorológicos utilizando rede lstm. In: UFA. **VIII Congresso Brasileiro de Energia Solar**. Fortaleza, 2020. Citado 2 vezes nas páginas 5 e 7.
- BEAR, M.; CONNORS, B.; PARADISO, M. **Neurociências: Desvendando o Sistema Nervoso**. Artmed Editora, 2002. ISBN 9788582714331. Disponível em: <<https://books.google.com.br/books?id=bywtDwAAQBAJ>>. Citado 2 vezes nas páginas 13 e 14.
- BROWNLEE, J. **Long short-term memory networks with python: develop sequence prediction models with deep learning**. Melbourne: Machine Learning Mastery, 2017. Citado 2 vezes nas páginas 18 e 45.
- BRUCE, A.; BRUCE, P. **Estatística Prática para Cientistas de Dados**. Rio de Janeiro: Alta Books, 2019. Citado na página 8.
- CHAI, T.; DRAXLER, R. R. Root mean square error (rmse) or mean absolute error (mae)?—arguments against avoiding rmse in the literature. **Geoscientific model development**, Copernicus GmbH, v. 7, n. 3, p. 1247–1250, 2014. Citado na página 24.
- ESTATIDADOS. **Redes Neurais Artificiais - Conceitos teóricos e práticos básicos - Matheus Pussaignolli**. Youtube, 2019. Acessado em: 08 de novembro de 2021. Disponível em: <<https://www.youtube.com/watch?v=FCRStdK9hRg&t=52s>>. Citado na página 13.
- FACELI, K. et al. **Inteligência artificial : uma abordagem de aprendizado de máquina**. 2. ed. Rio de Janeiro: LTC, 2021. Citado 4 vezes nas páginas 1, 13, 16 e 18.
- FRACCANABBIA, N.; MARIANI, V. C. Avaliando aprendizado de máquina na previsão de curto prazo de séries temporais de energia solar. **Revista Brasileira de Computação Aplicada**, v. 13, n. 2, p. 105–112, 2021. Citado na página 2.
- GÉRON, A. **Mãos à Obra: Aprendizado de Máquina com Scikit-Learn, Keras TensorFlow**. Rio de Janeiro: Alta Books, 2021. Citado 4 vezes nas páginas 10, 17, 18 e 24.
- GUPTA, C. B.; GUTTMAN, I. **Estatística e probabilidade com aplicações para engenheiros e cientistas**. Rio de Janeiro: Grupo Gen-LTC, 2017. 2188 p. Citado 2 vezes nas páginas 9 e 11.
- HAYKIN, S. **Redes neurais: princípios e prática**. Porto Alegre: Bookman Editora, 2007. Citado 4 vezes nas páginas 12, 13, 14 e 15.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural computation**, MIT Press, v. 9, n. 8, p. 1735–1780, 1997. Citado na página 19.

IEA. **Renewables 2020**. 2020. Acessado em: 11 de outubro de 2021. Disponível em: <<https://www.iea.org/reports/renewables-2020>>. Citado na página 5.

IEA. **Snapshot of Global PV Markets 2021**. 2021. Acessado em: 30 de outubro de 2021. Disponível em: <https://iea-pvps.org/wp-content/uploads/2021/04/IEA_PVPS_Snapshot_2021-V3.pdf>. Citado na página 6.

KANNAL, A. **Solar Power Generation Data: Solar power generation and sensor data for two power plants**. Kaggle, 2020. Acessado em: 5 de outubro de 2021. Disponível em: <<https://www.kaggle.com/anikannal/solar-power-generation-data>>. Citado na página 25.

KINGMA, D. P.; BA, J. **Adam: a method for stochastic Optimization**. 2017. Citado na página 45.

KOPP, G.; LEAN, J. L. A new, lower value of total solar irradiance: Evidence and climate significance. **Geophysical Research Letters**, Wiley Online Library, v. 38, n. 1, 2011. Citado na página 26.

LABS, W. **Redes Neurais Desmitificadas [Parte 1: Dados e Arquitetura]**. Youtube, 2014. Acessado em: 08 de novembro de 2021. Disponível em: <<https://www.youtube.com/watch?v=bxe2T-V8XR8>>. Citado na página 14.

LIU, L. et al. Forecasting power output of photovoltaic system using a bp network method. **Energy Procedia**, Elsevier, v. 142, p. 780–786, 2017. Citado na página 7.

MILOCA, S. A.; CONEJO, P. D. Multicolinearidade em modelos de regressão. **Semana acadêmica da matemática**, n. 22, p. 25–34, 2008. Citado na página 10.

NAVIDI, W. **Probabilidade e estatística para ciências exatas**. Nova Iorque: McGraw Hill Brasil, 2012. Citado na página 12.

OLAH, C. **Understanding LSTM Networks**. 2015. Acessado em: 14 de novembro de 2021. Disponível em: <<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>>. Citado 6 vezes nas páginas 18, 19, 20, 21, 22 e 23.

RITCHIE, H.; ROSER, M. **Energy Production and Consumption**. 2020. Acessado em: 11 de outubro de 2021. Disponível em: <<https://ourworldindata.org/energy-production-consumption>>. Citado na página 2.

ROHRER, B. **Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM)**. Youtube, 2017. Acessado em: 14 de novembro de 2021. Disponível em: <<https://www.youtube.com/watch?v=WCUNPb-5EYI>>. Citado na página 21.

SANTOS, S. S. d. Previsão da produção de uma usina fotovoltaica usando redes neurais artificiais. Universidade Federal de Goiás, 2019. Citado na página 76.

SICSÚ, A.; DANA, S. **ESTATÍSTICA APLICADA: ANÁLISE EXPLORATÓRIA DE DADOS**. [S.l.]: Saraiva, 2012. Citado na página 10.

SU, D.; BATZELIS, E.; PAL, B. Machine learning algorithms in forecasting of photovoltaic power generation. In: IEEE. **2019 International Conference on Smart Energy Systems and Technologies (SEST)**. [S.l.], 2019. p. 1–6. Citado 2 vezes nas páginas 6 e 7.

TECH, D. **Coeficiente de determinação R2 (machine learning)**. Youtube, 2019. Acessado em: 11 de janeiro de 2022. Disponível em: <<https://www.youtube.com/watch?v=wDJmH3JFqc>>. Citado na página 23.

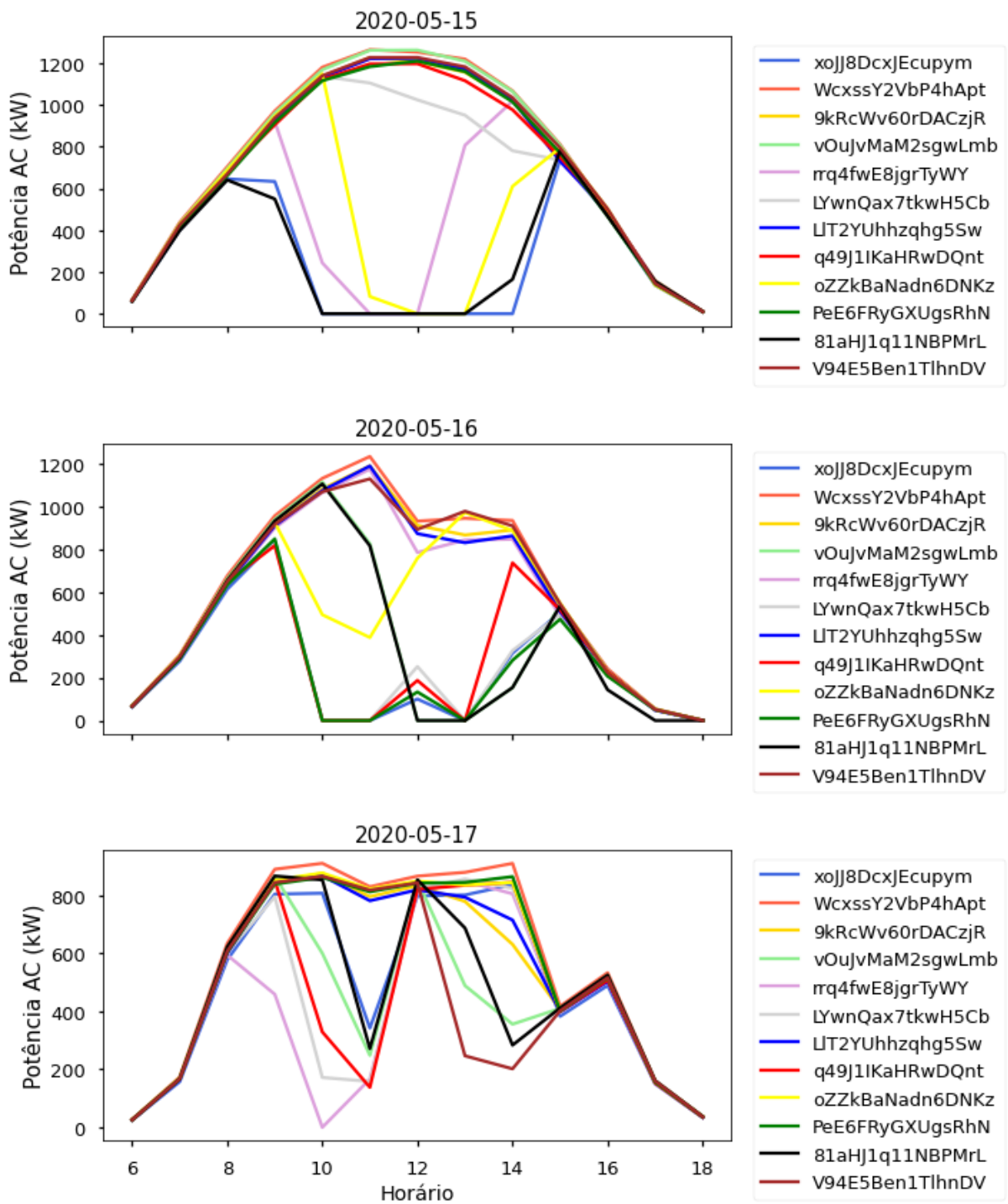
TORRES, S. et al. Previsão do preço de ações brasileiras utilizando redes neurais artificiais. Universidade Presbiteriana Mackenzie, 2021. Citado na página 7.

VIAN, Â. et al. **Energia Solar: Fundamentos Tecnologia e Aplicações**. São Paulo: Editora Blücher, 2021. Citado 2 vezes nas páginas 1 e 2.

Apêndices

APÊNDICE A – Potência AC gerada para cada um dos 34 dias, sem tratamento para nulos

Figura 61 – Potência AC gerada nos dias 15/05/20 a 17/05/20 sem tratamento.



Fonte: (Autor, 2022)

Figura 62 – Potência AC gerada nos dias 18/05/20 a 21/05/20 sem tratamento.

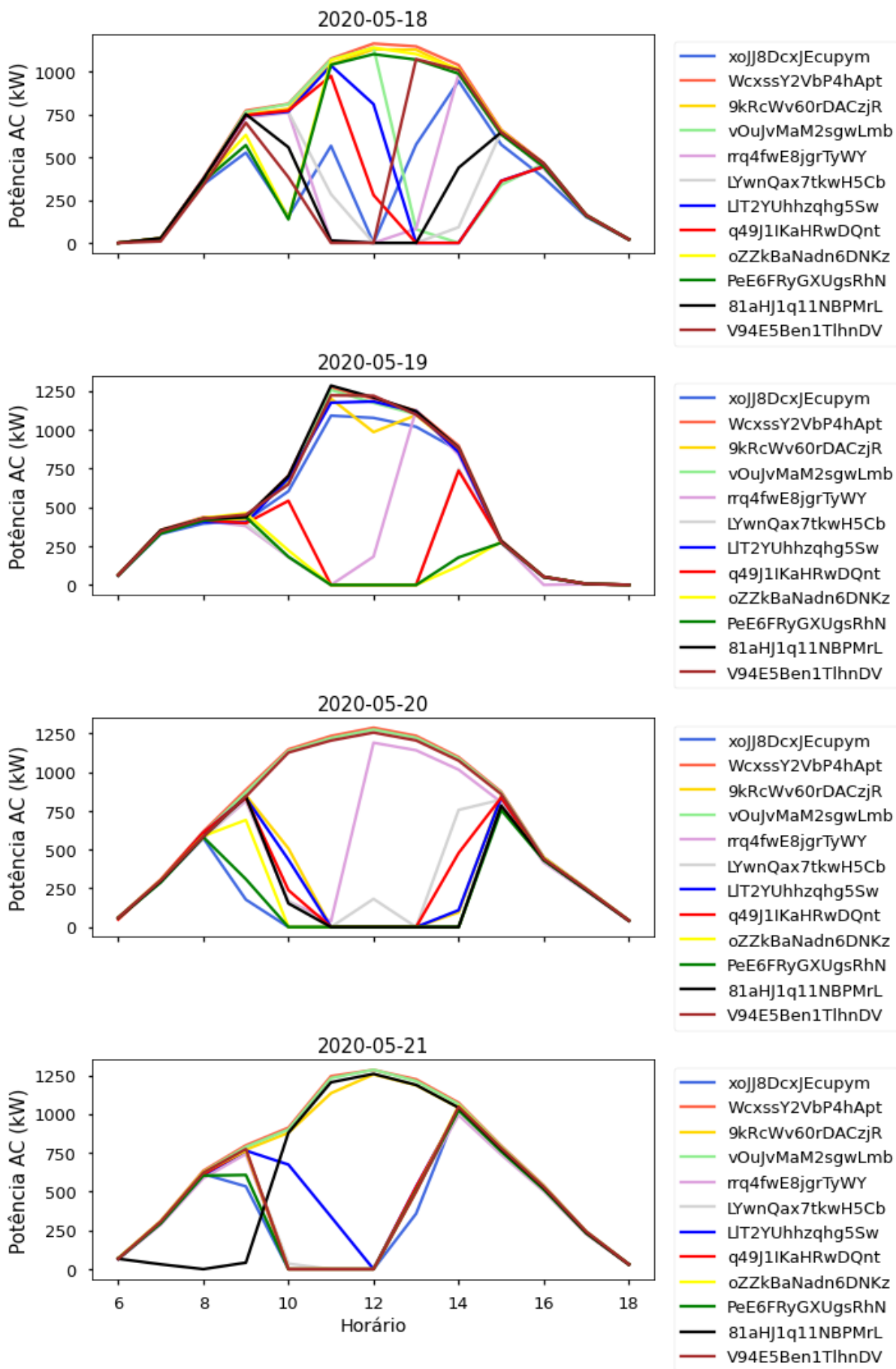


Figura 63 – Potência AC gerada nos dias 22/05/20 a 25/05/20 sem tratamento.

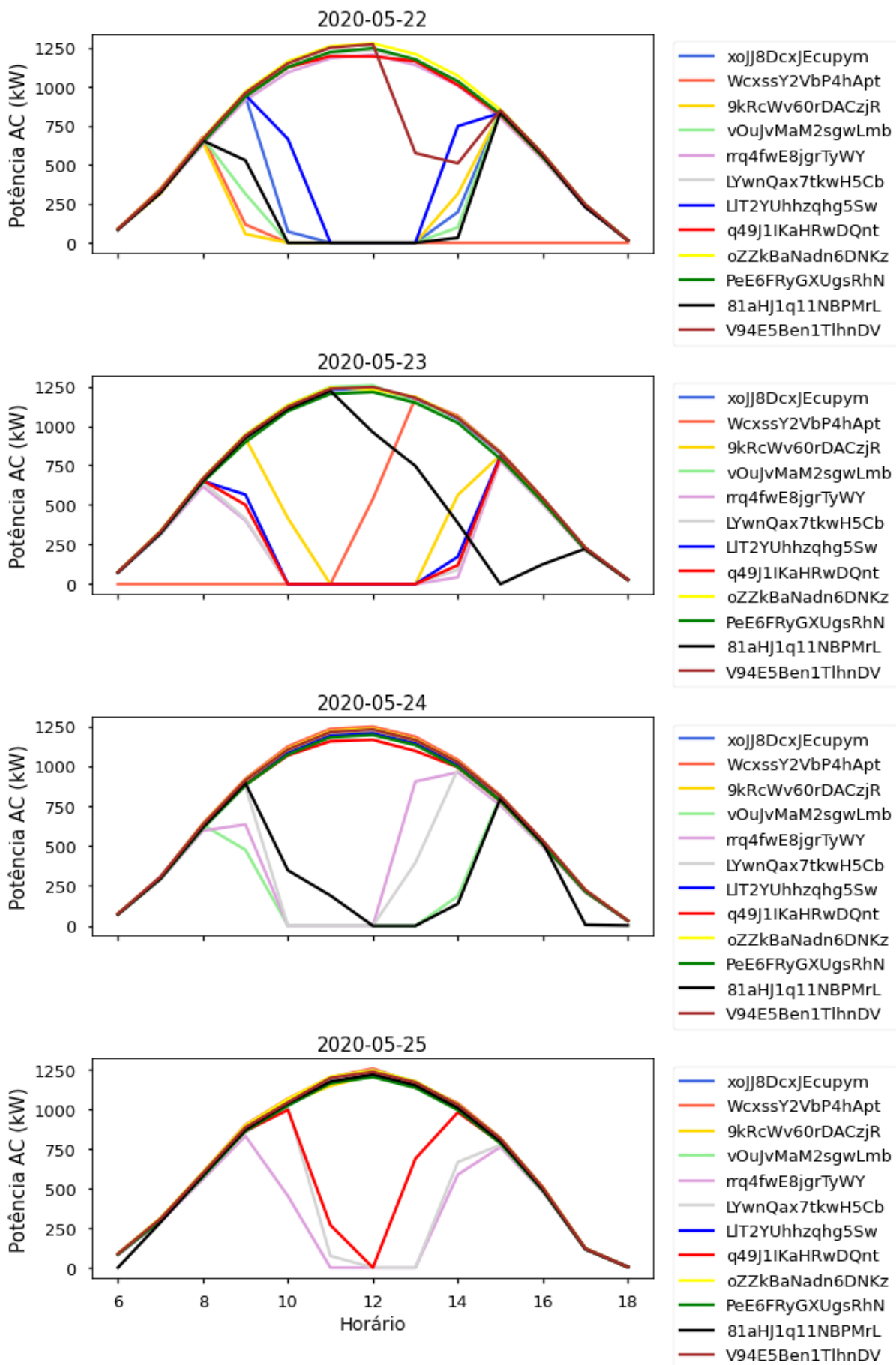


Figura 64 – Potência AC gerada nos dias 26/05/20 a 29/05/20 sem tratamento.

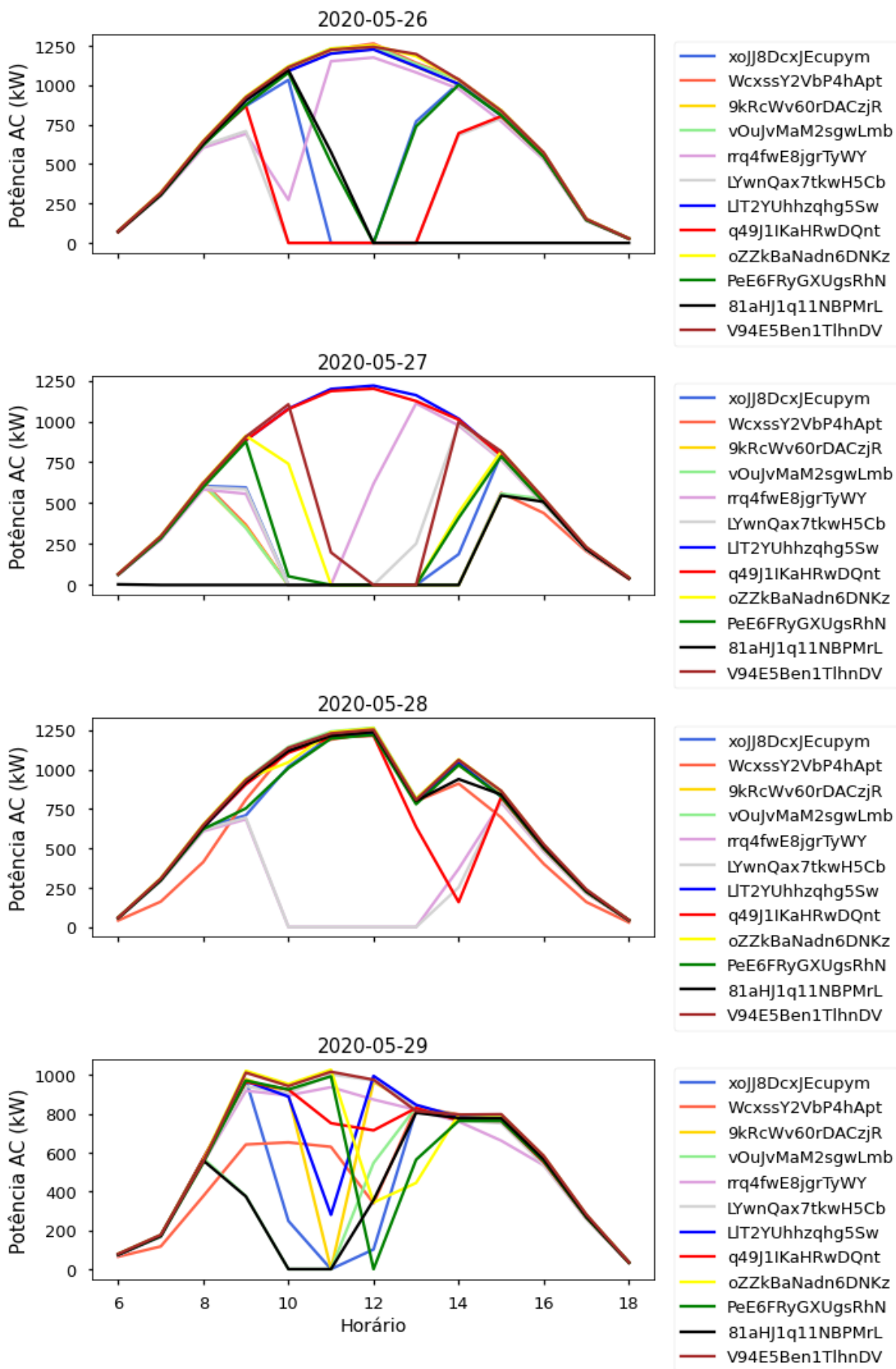


Figura 65 – Potência AC gerada nos dias 30/05/20 a 02/06/20 sem tratamento.

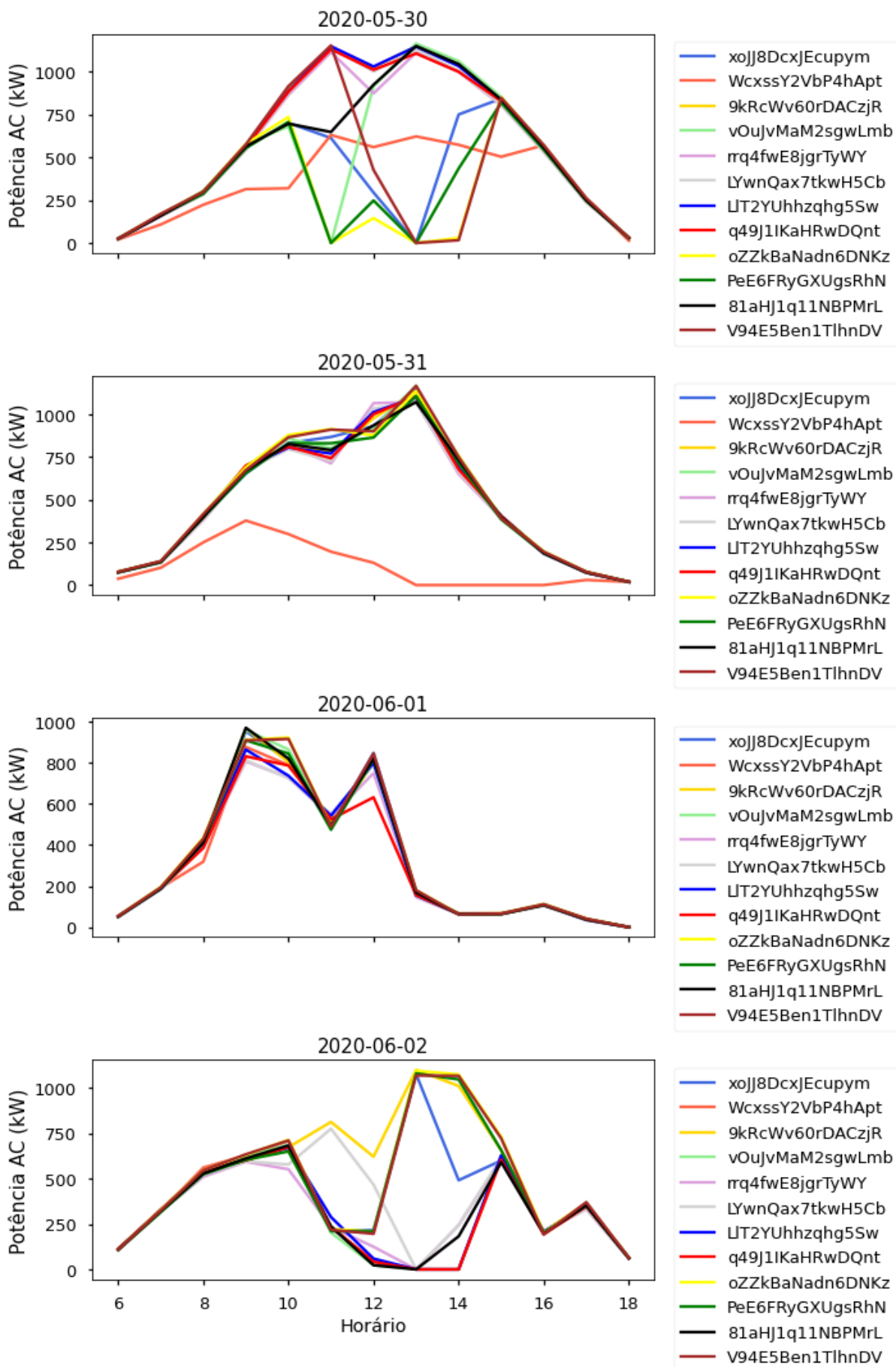


Figura 66 – Potência AC gerada nos dias 03/06/20 a 06/06/20 sem tratamento.

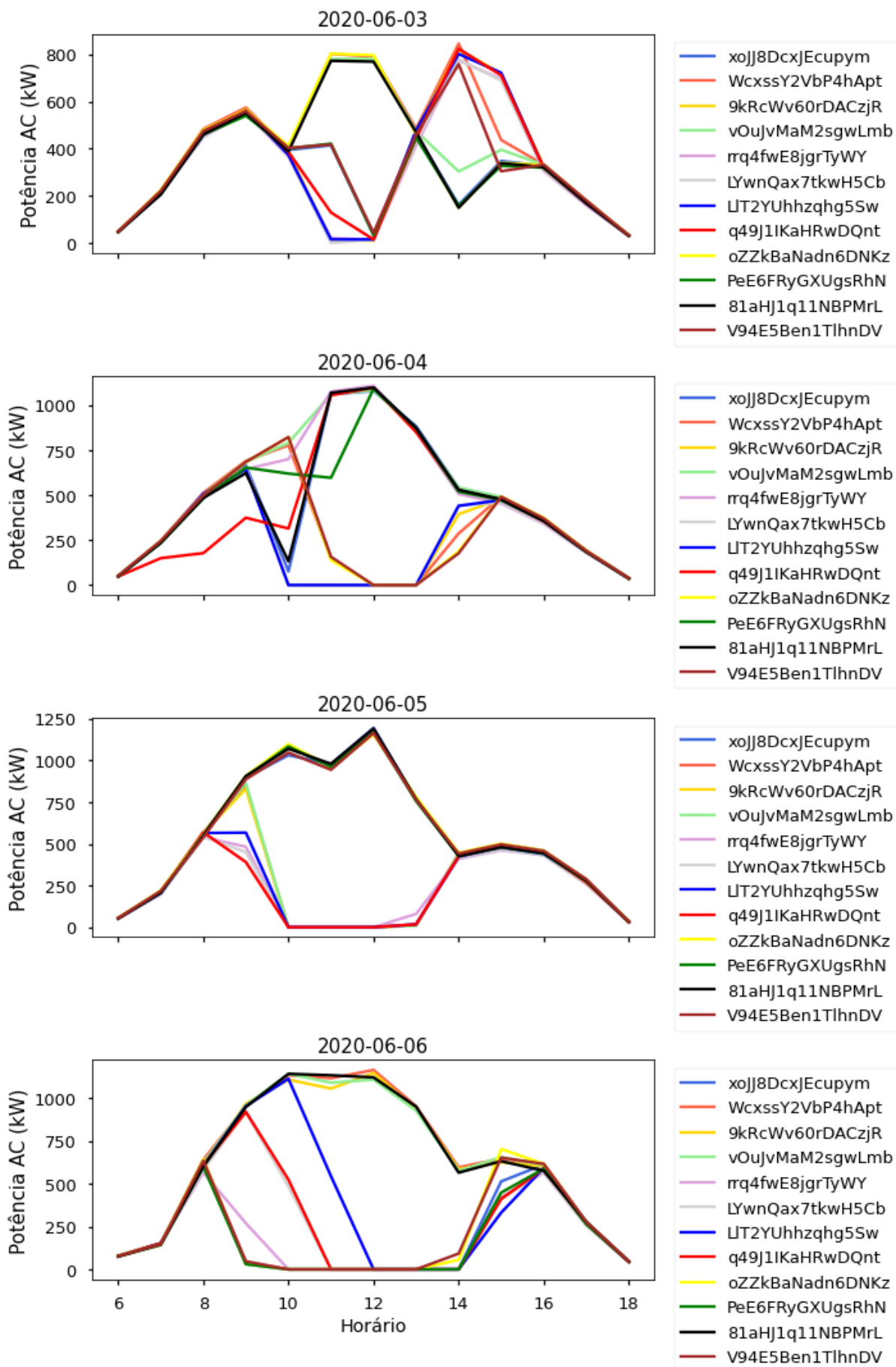


Figura 67 – Potência AC gerada nos dias 07/06/20 a 10/06/20 sem tratamento.

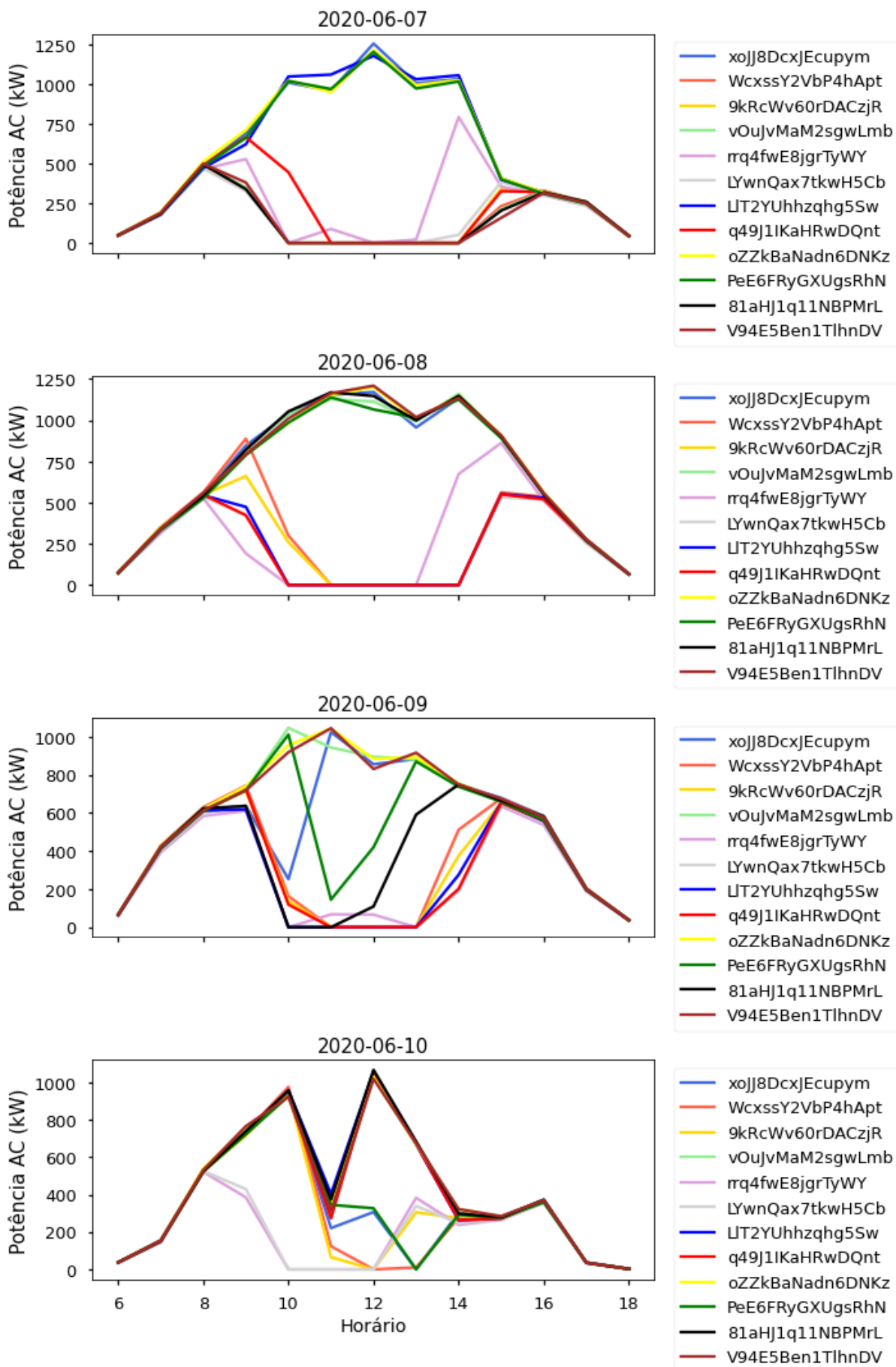


Figura 68 – Potência AC gerada nos dias 11/06/20 a 14/06/20 sem tratamento.

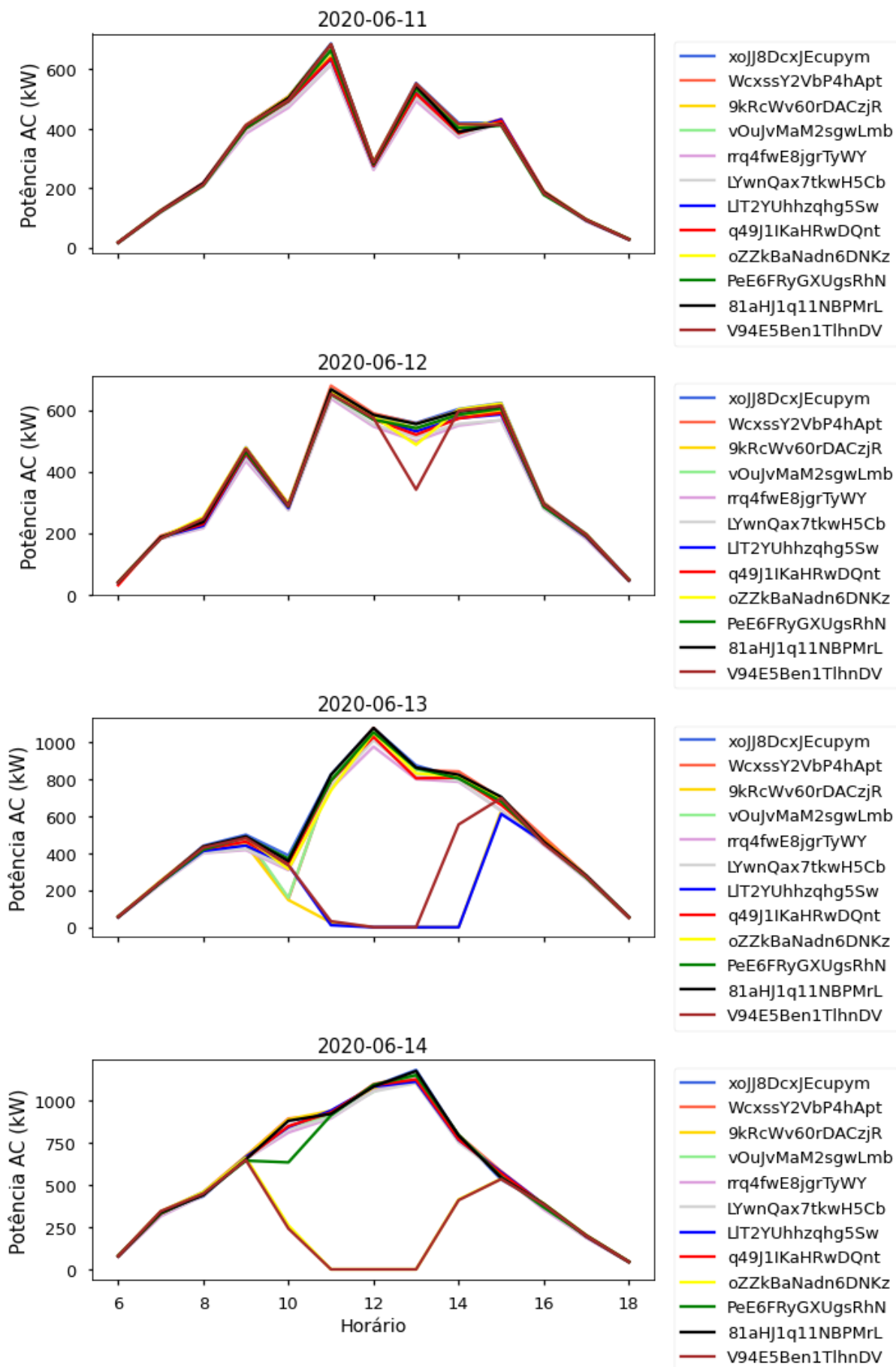
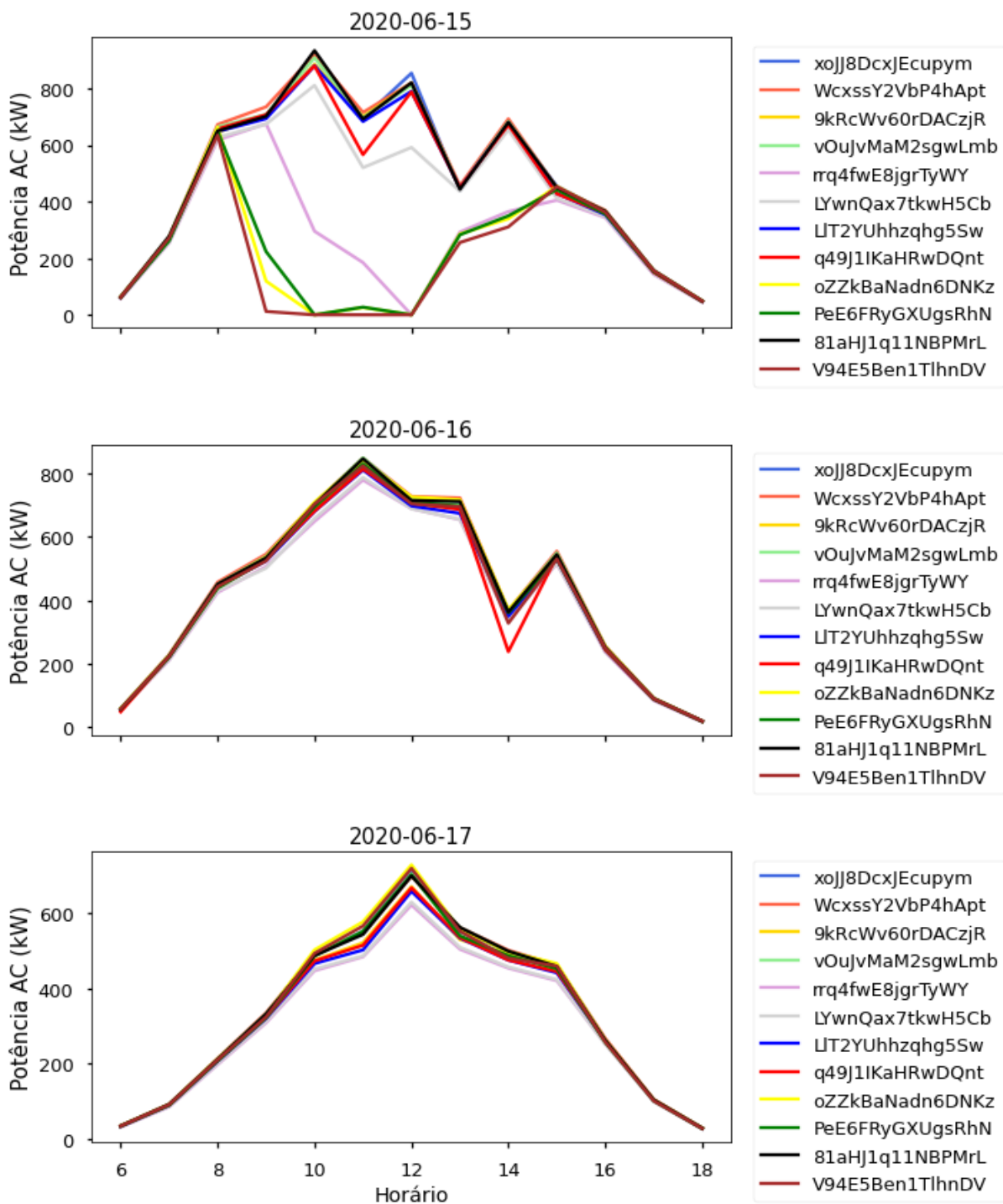


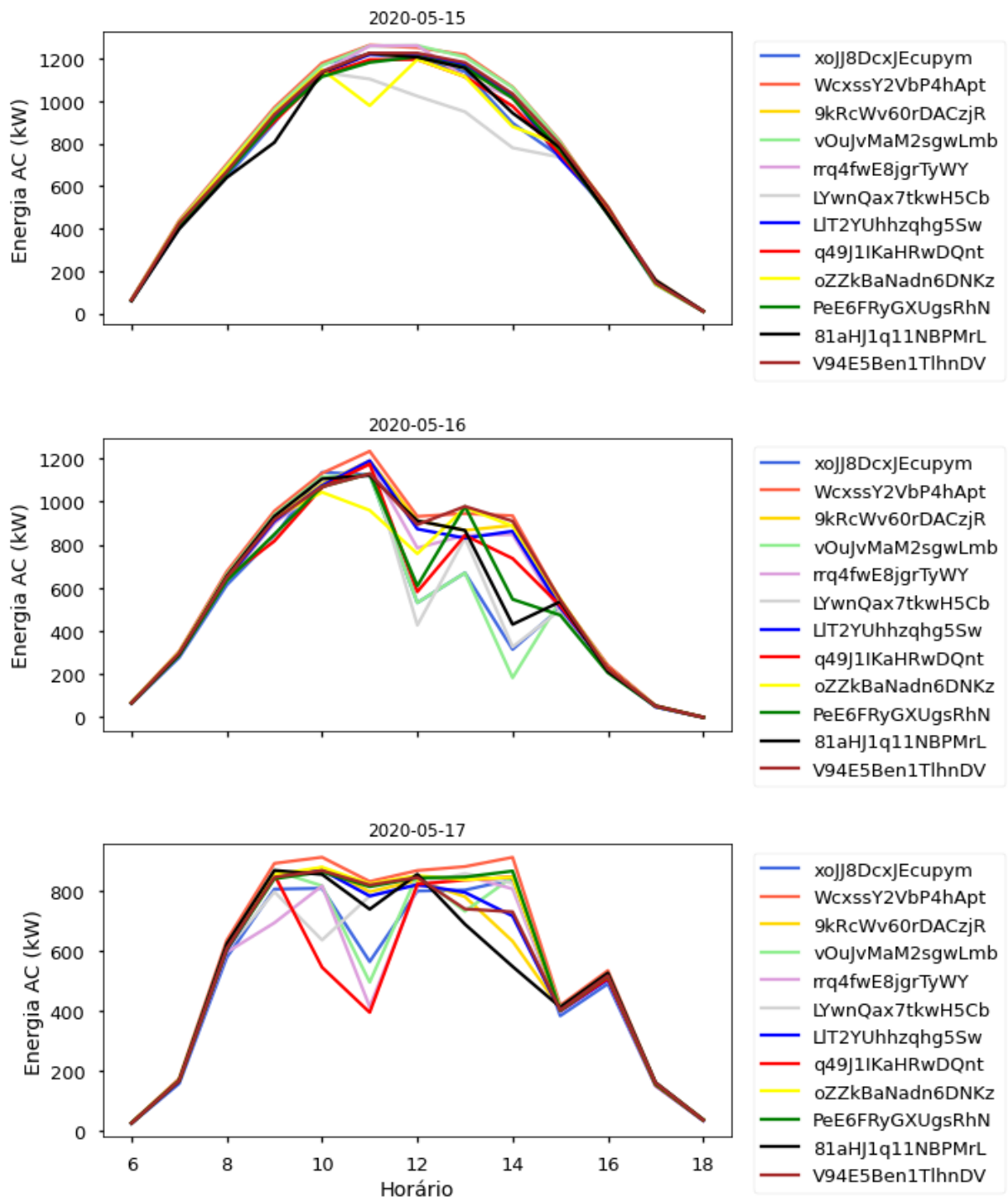
Figura 69 – Potência AC gerada nos dias 15/06/20 a 17/06/20 sem tratamento.



Fonte: (Autor, 2022)

APÊNDICE B – Potência AC gerada para cada um dos 34 dias, com tratamento para nulos

Figura 70 – Potência AC gerada nos dias 15/05/20 a 17/05/20 com tratamento.



Fonte: (Autor, 2022)

Figura 71 – Potência AC gerada nos dias 18/05/20 a 21/05/20 com tratamento.

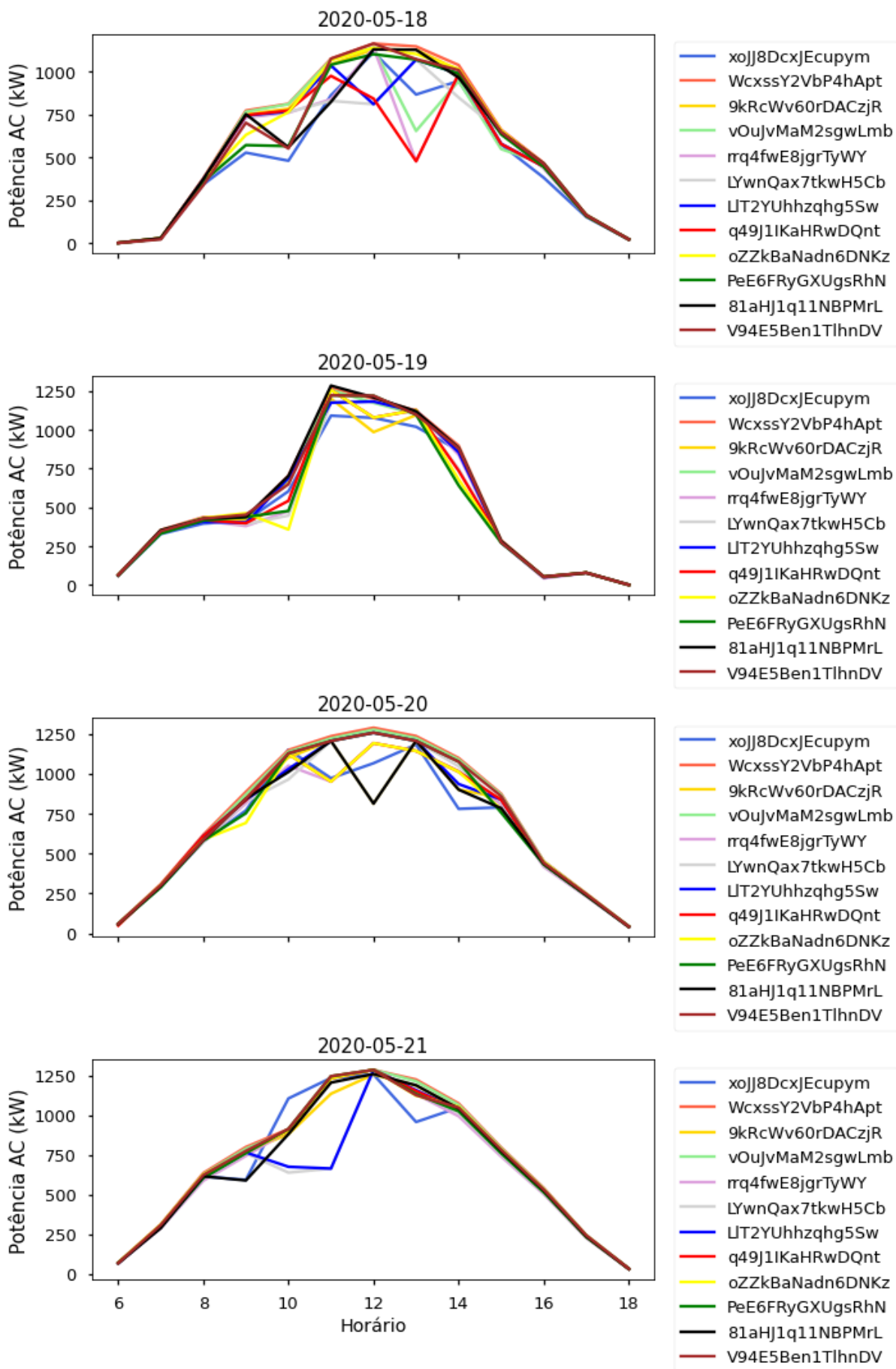


Figura 72 – Potência AC gerada nos dias 22/05/20 a 25/05/20 com tratamento.

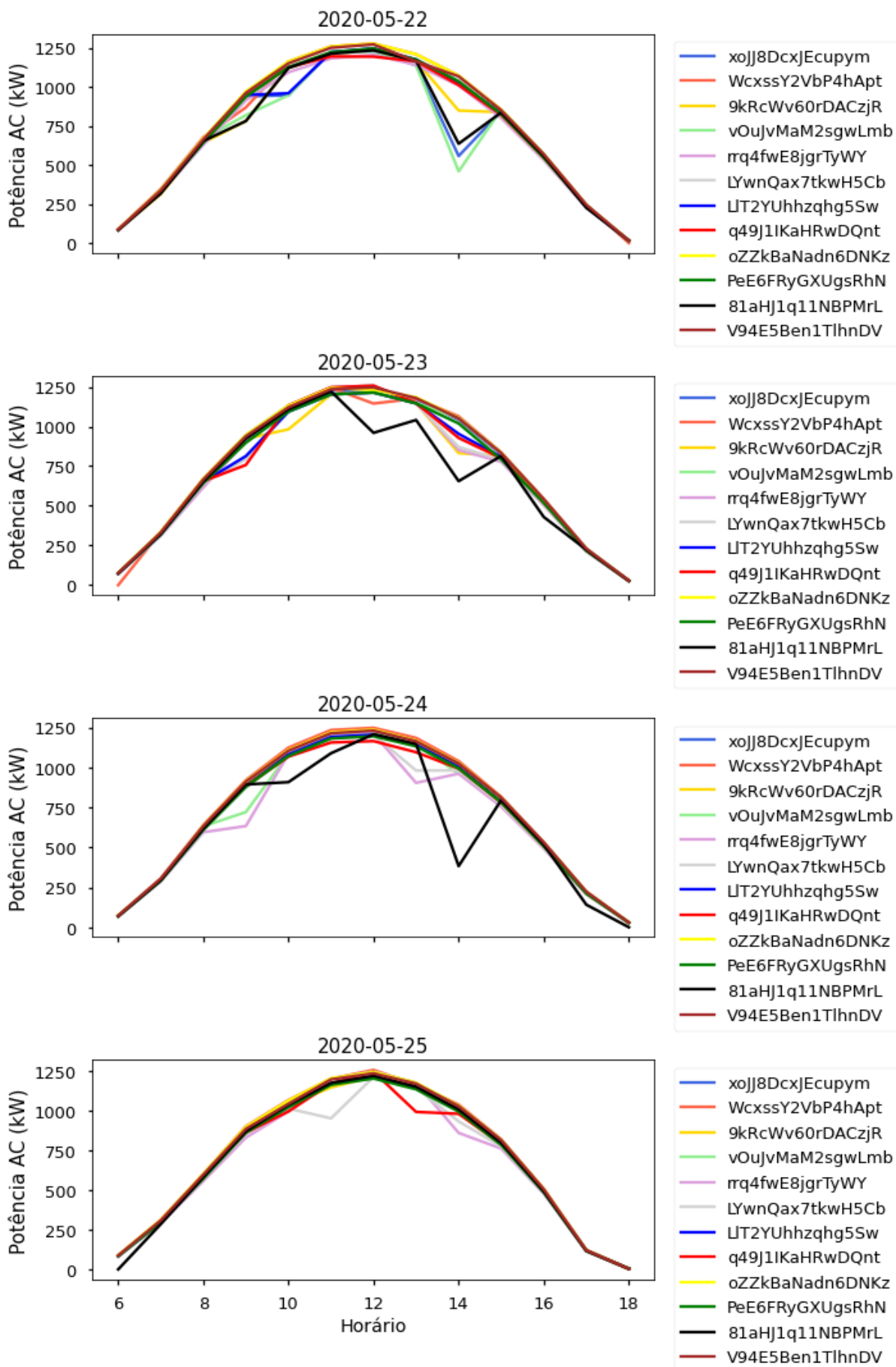


Figura 73 – Potência AC gerada nos dias 26/05/20 a 29/05/20 com tratamento.

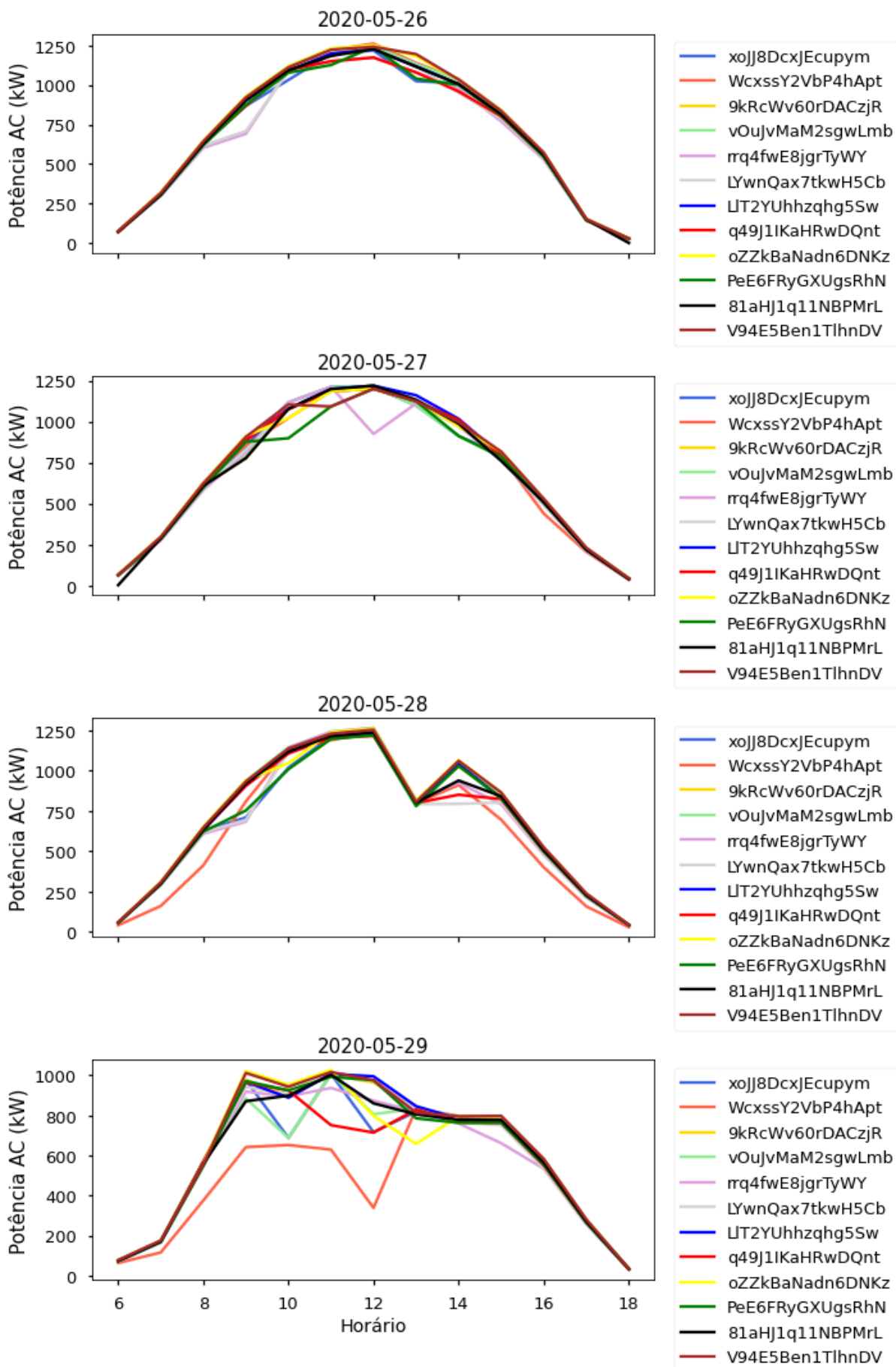


Figura 74 – Potência AC gerada nos dias 30/05/20 a 02/06/20 com tratamento.

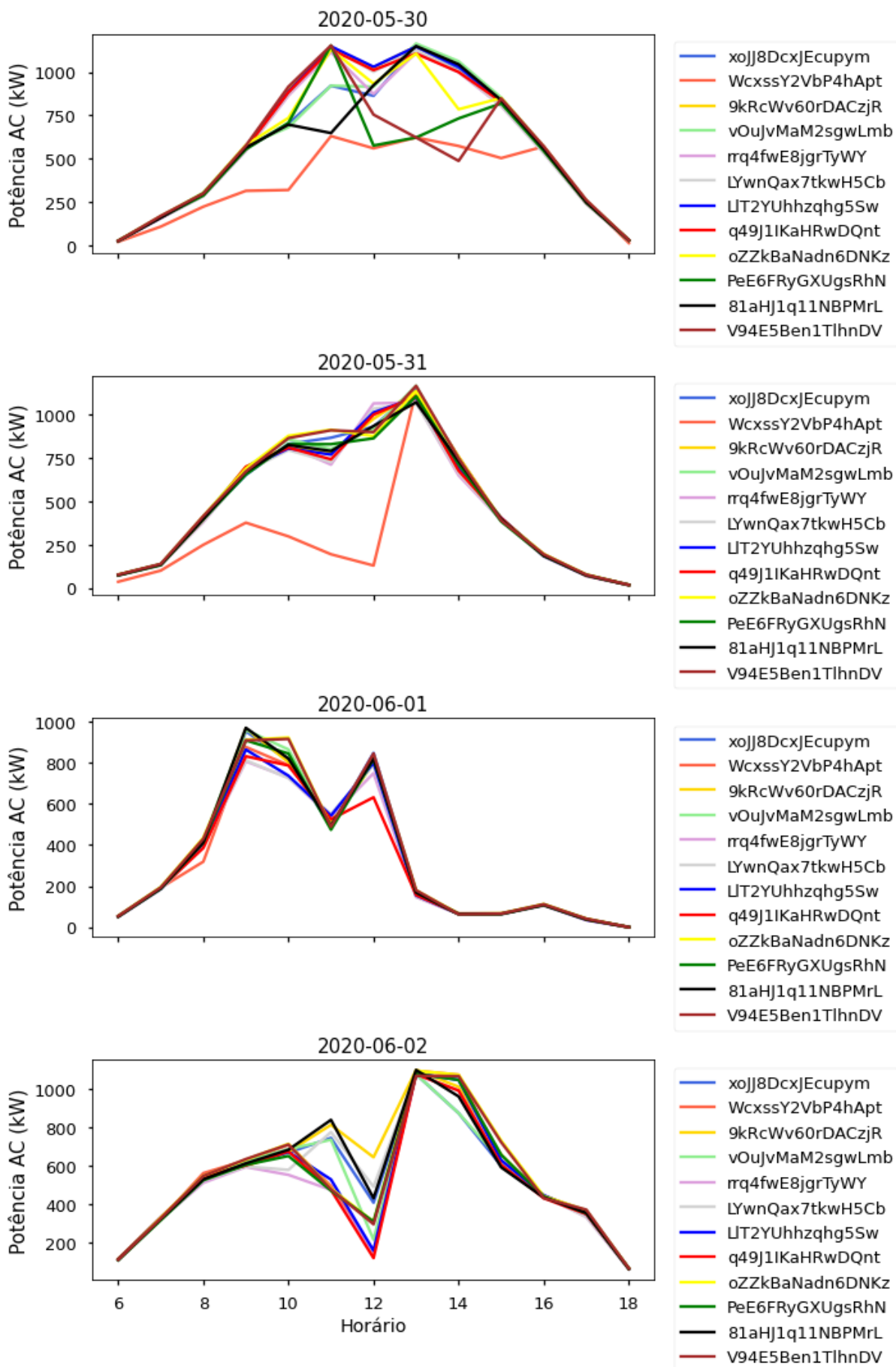


Figura 75 – Potência AC gerada nos dias 03/06/20 a 06/06/20 com tratamento.

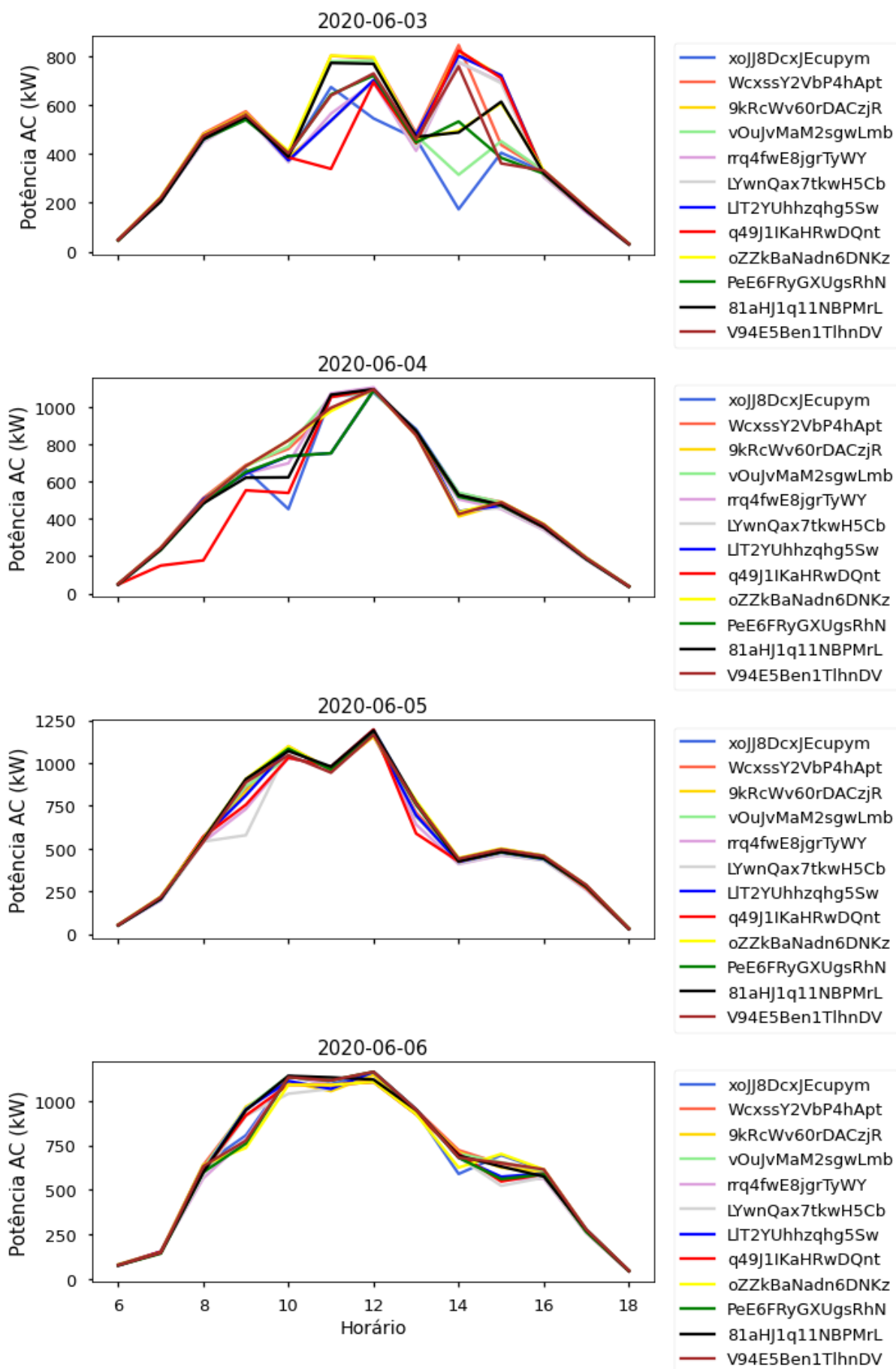


Figura 76 – Potência AC gerada nos dias 07/06/20 a 10/06/20 com tratamento.

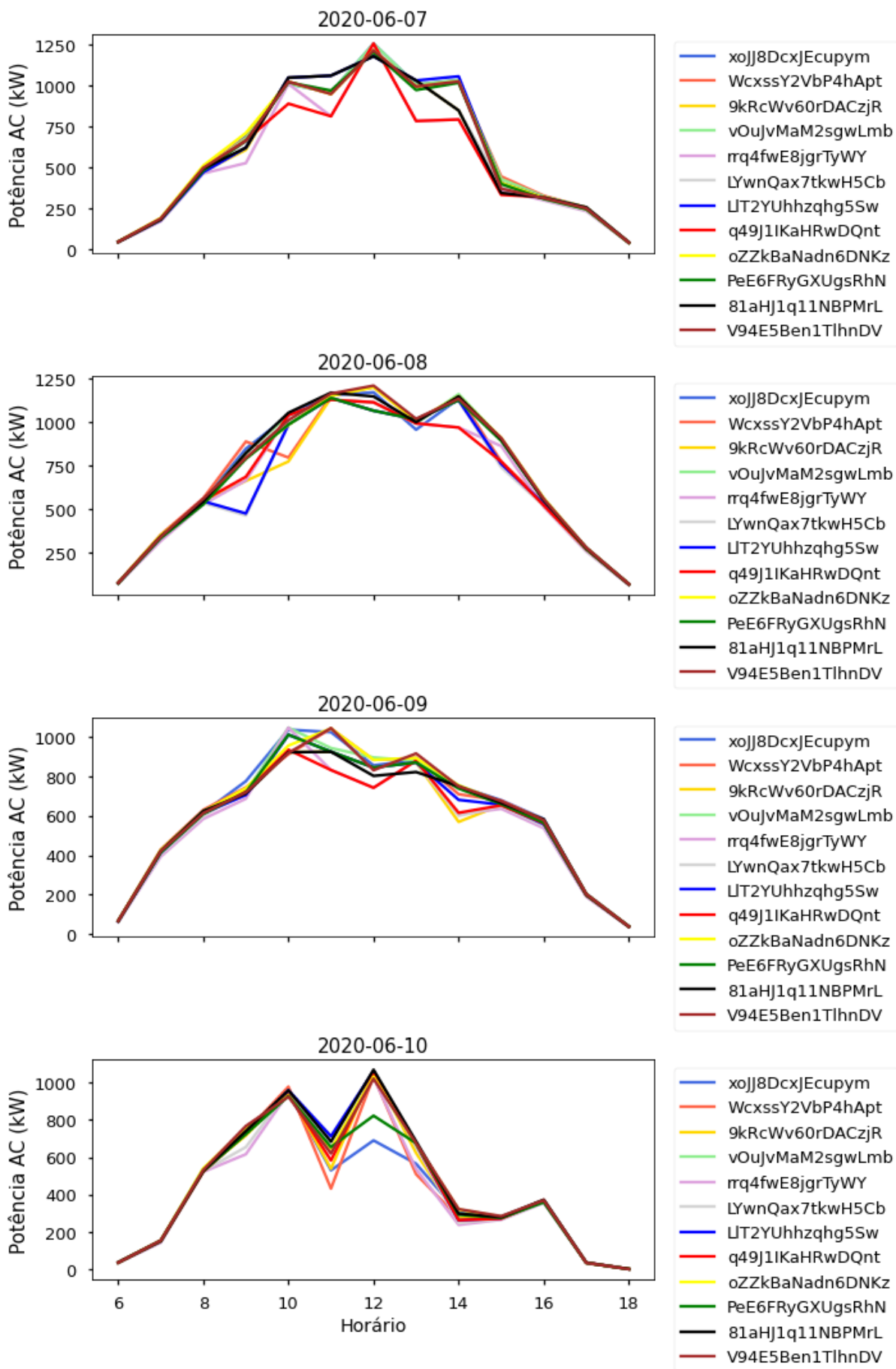


Figura 77 – Potência AC gerada nos dias 11/06/20 a 14/06/20 com tratamento.

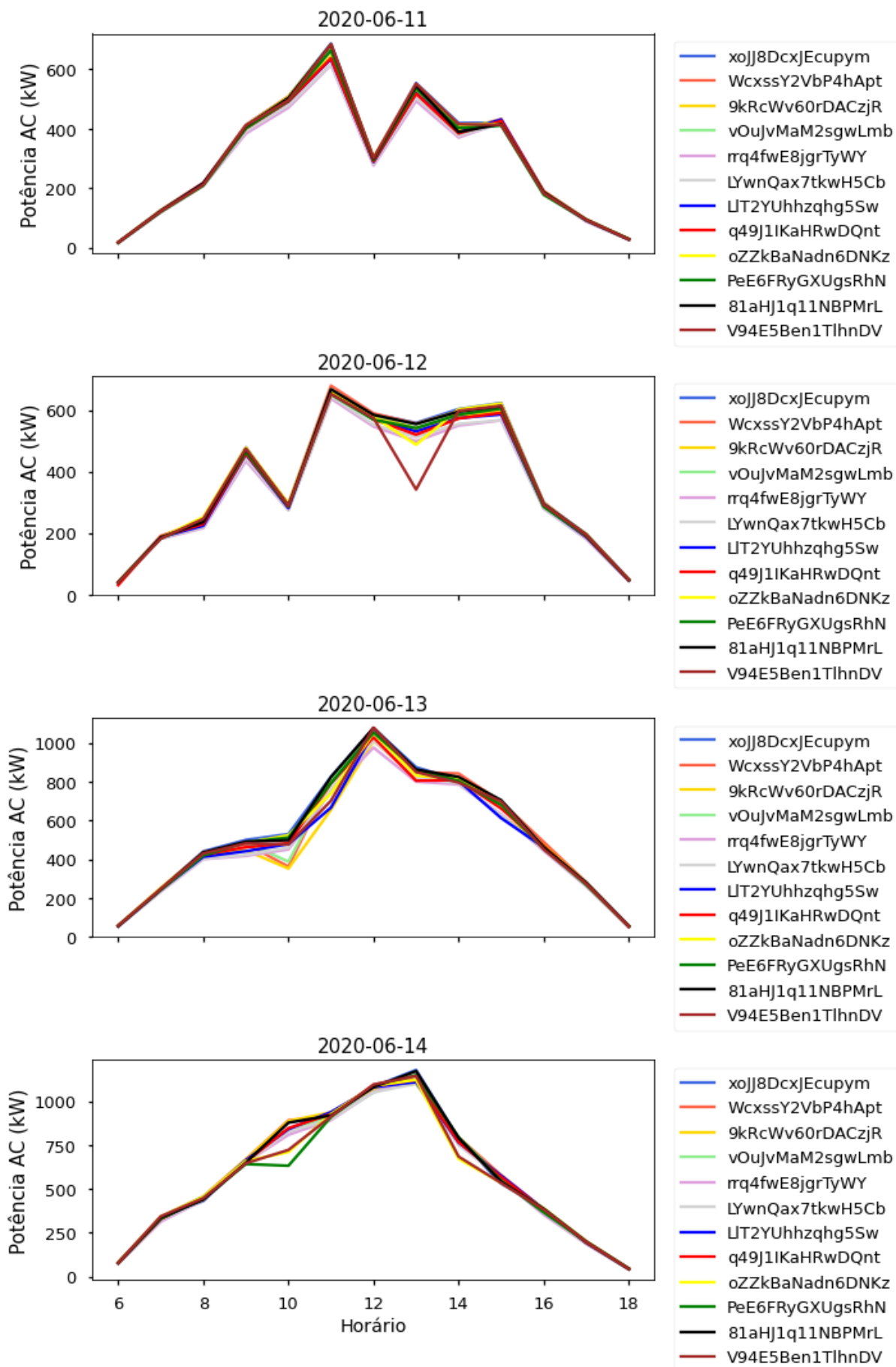
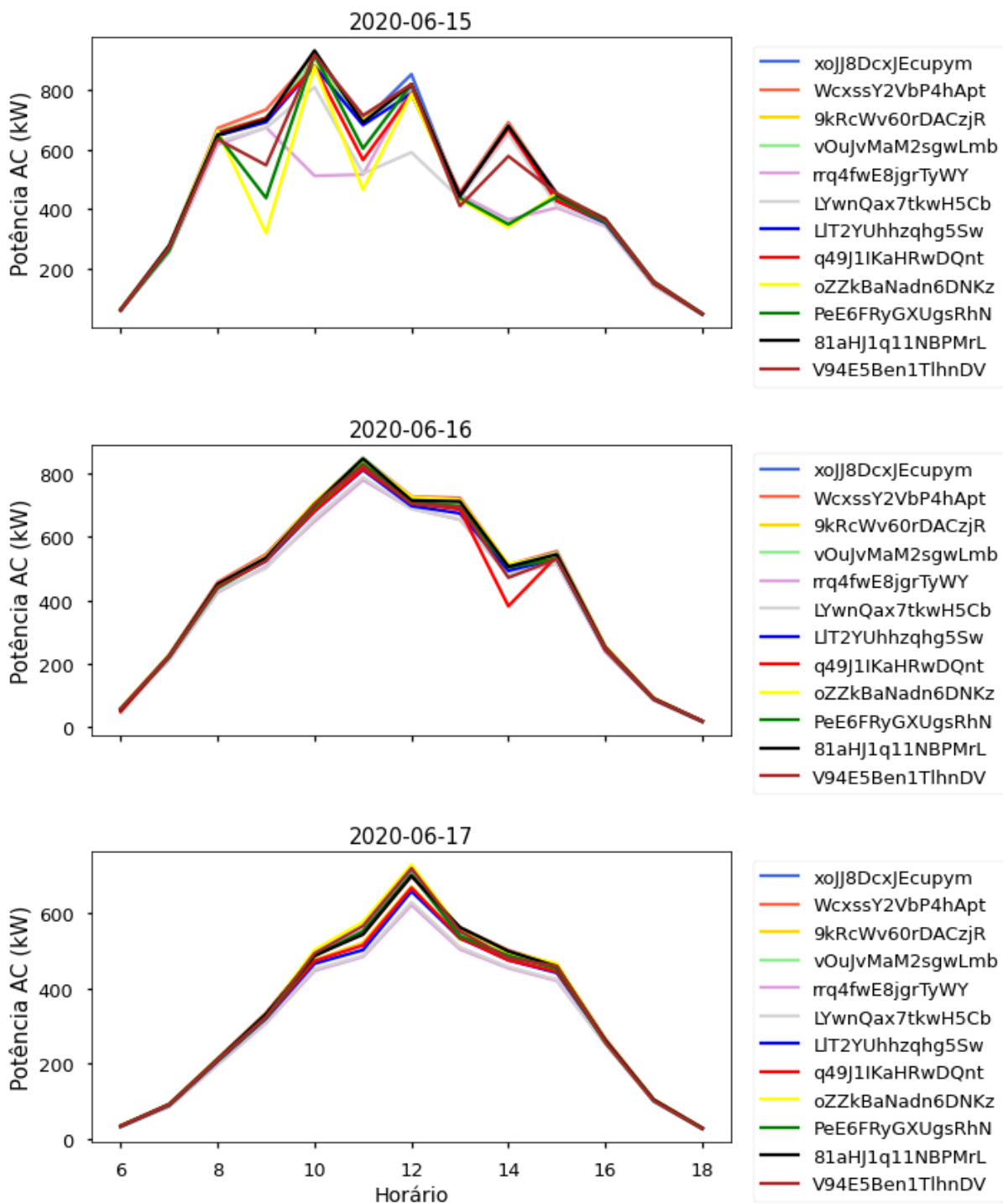


Figura 78 – Potência AC gerada nos dias 15/06/20 a 17/06/20 com tratamento.



Fonte: (Autor, 2022)