UNIVERSIDADE DE SÃO PAULO
Instituto de Ciências Matemáticas e de Computação

Algoritmo Previsor de Carregamento de Transformadores baseado em Técnicas de Inteligência Artificial

Filipe Rodrigues Lopes

Monografia - MBA em Inteligência Artificial e Big Data

ICMC USP SÃO CARLOS

Filipe Rodrigues Lopes

Algoritmo Previsor de Carregamento de Transformadores baseado em Técnicas de Inteligência Artificial

Monografia apresentada ao Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, como parte dos requisitos para obtenção do título de Especialista em Inteligência Artificial e Big Data.

Área de concentração: Inteligência Artificial

Orientador: Prof. Dr. Leandro Franco de Souza

Versão original

São Carlos

2023

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi, ICMC/USP, com os dados fornecidos pelo(a) autor(a)

Lopes, Filipe Rodrigues
L864a Algoritmo Previsor de Carregamento de Transformadores baseado em Técnicas de Inteligência Artificial / Filipe Rodrigues Lopes ; orientador Leandro Franco de Souza. – São Carlos, 2023.
127 p. : il. (algumas color.) ; 30 cm.

Monografia (MBA em Inteligência Artificial e Big Data) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2023.

1. Inteligência Artificial. 2. Séries Temporais. 3. Modelos de Regressão. 4. Redes Neurais. I. de Souza, Leandro Franco, orient. II. Título.

Filipe Rodrigues Lopes

Algoritmo Previsor de Carregamento de Transformadores baseado em Técnicas de Inteligência Artificial

Monograph presented to the Departamento de Ciências de Computação do Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo - ICMC/USP, as part of the requirements for obtaining the title of Specialist in Artificial Intelligence and Big Data.

Concentration area: Artificial Intelligence

Advisor: Prof. Dr. Leandro Franco de Souza

Original version

São Carlos

2023

Este trabalho é dedicado aos meus pais,
Edson e Ester, por toda educação, inspiração e incentivo;
À minha esposa e filhos, Raíta, Daniel e Pedro,
pelo indubitável amor, carinho e compreensão, sempre presente.
E à Deus, razão do existir, porque tudo é dEle, por Ele e para Ele.

AGRADECIMENTOS

À Eletrobras Eletronorte, na pessoa do Dr. Antonio Pardauil, por todos os recursos para a realização deste trabalho.

À Dr[a] Mônica Teixeira e ao Eng. Maurício Iryoda, pela oportunidade, reconhecimento e, sobretudo, pela amizade ao longo de toda caminhada até aqui na Eletronorte.

Aos amigos da OOSP, pelo companheirismo, colaboração, suporte e incentivo. Que este trabalho seja útil para o nosso trabalho.

Aos companheiros de curso, os Engs. Claudio Cabral, Felipe Castelar, Kenneth Mendonça, Newton Teixeira e Walmor Gomes, sem os quais seria muito mais difícil.

Ao professor Dr. Leandro Franco de Souza e à professora Dr[a] Solange Oliveira Rezende por todo ensino ao longo desse tempo, inclusive o acadêmico.

"Lembrem-se das coisas passadas, das coisas da antiguidade: que eu sou Deus, e não há outro; eu sou Deus, e não há outro semelhante a mim. Desde o princípio anuncio o que há de acontecer e desde a antiguidade revelo as coisas que ainda não sucederam"
(Isaias 46:9-10 NAA)

RESUMO

Lopes, F. R. **Algoritmo Previsor de Carregamento de Transformadores baseado em Técnicas de Inteligência Artificial**. 2023. 127p. Monografia (MBA em Inteligência Artificial e Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2023.

Este trabalho estuda diferentes técnicas de inteligência artificial, buscando a melhor técnica para um algoritmo preditor de série temporal, capaz de prever a carga nos transformadores da Eletrobras Eletronorte.

Os dados foram obtidos pela Eletronorte para os transformadores da SE Porto Velho 230/69/13,8 kV 4x100 MVA, correspondendo a 11 meses de medição. Após tratamento, adicionaram-se atributos de clima e características diárias. A análise exploratória dos dados mostrou baixa correlação entre os atributos adicionados e a carga, e que os dados de carga são uma série temporal estacionária, autocorrelacionada, mas não normal. Essas características subsidiaram a escolha dos modelos: Regressão Tradicionais, Support Vector Machine, Árvore de Decisão como Random Forest e XGBoost, e Redes Neurais, incluindo Convolucionais e Recorrentes.

Os resultados iniciais indicaram ótimo desempenho, com destaque para o Random Forest, que alcançou R2 de 0.984 e MSE de 11.23. Análises complementares foram feitas, revelando a importância de atributos que descrevem a variação temporal dos instantes imediatamente anterior em todos os modelos, exceto nas Redes Neurais, onde esses atributos podem ser suprimidos sem prejudicar o resultado.

Após reavaliação da arquitetura das redes neurais e do aperfeiçoamento do conjunto de dados, foi possível melhorar o desempenho computacional destes modelos, com destaque para o LSTM e a GRU, que alcançaram valores de R2 e MSE melhores que o Random Forest.

Palavras-chave: Predição de Carga. Séries Temporais. Regressão Linear e Polinomial. *Support Vector Machine*. Árvores de Decisão. Redes Neurais.

ABSTRACT

Lopes, F. R. **Transformer Loading Predictor Algorithm based in Artificial Intelligence Techniques**. 2023. 127p. Monograph (MBA in Artificial Intelligence and Big Data) - Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2023.

This study investigates various artificial intelligence techniques, aiming to identify the optimal approach for a time series forecasting algorithm, designed to predict the load on transformers of Eletrobras Eletronorte.

The data were sourced from Eletronorte for the transformers at SE Porto Velho 230/69/13.8 kV 4x100 MVA, spanning an 11-month measurement period. Following data preprocessing, climatic attributes and daily features were incorporated. The exploratory data analysis revealed a low correlation between the added attributes and the load, noting that the load data constitutes a stationary and autocorrelated time series, albeit not normally distributed. These insights informed the selection of models: Traditional Regression, Support Vector Machine, Decision Trees such as Random Forest and XGBoost, and Neural Networks, encompassing both Convolutional and Recurrent types.

Preliminary results indicated excellent performance, with Random Forest standing out, achieving an R2 of 0.984 and an MSE of 11.23. Further analyses underscored the significance of attributes detailing the temporal variation of the immediate previous instances in all models, except for Neural Networks, where these attributes can be omitted without compromising results.

Upon reevaluation of the neural network architecture and data set refinement, computational performance enhancement of these models was observed. Notably, LSTM and GRU surpassed the R2 and MSE values achieved by Random Forest.

Keywords: Power Load Predictor. Time Series. Linear and Polynomial Regression. Support Vector Machine. Decision Trees. Neural Networks.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

ANEEL	Agência Nacional de Energia Elétrica
ACF	Função de Autocorrelação
ADF	Teste de Dickey-Fuller Aumentado
AED	Análise Exploratória de Dados
AR	*Autoregressive Model* ou Modelo Autoregressivo
ARIMA	*Autoregressive Integrated Moving Average Model* ou Modelo Autoregressivo Integrado de Média Móvel
ARMA	*Autoregressive Moving Average Model* ou Modelo Autoregressivo de Média Móvel
CEPEL	Centro de Pesquisas de Energia Elétrica
CNN	Redes Neurais Convolucionais
DW	Teste de Durbin-Watson
EUNITE	*European Network on Intelligent Technologies for Smart Adaptive Systems*
GRU	*Gated Recurrent Unit*
IA	Inteligência Artificial
kV	quilovolt
LSTM	*Long Short-Term Memory*
LTLF	*Long-Term Load Forecaster* ou Previsor de carga de longo-prazo
MA	Moving Average Models ou Modelo de Média Móvel
ML	*Machine Learning*
MLP	*Multi-layer Perceptron*
MSE	Erro Médio Quadrático
MTLF	*Mid-Term Load Forecaster* ou Previsor de carga de médio-prazo
MVA	Megavolt-Ampere

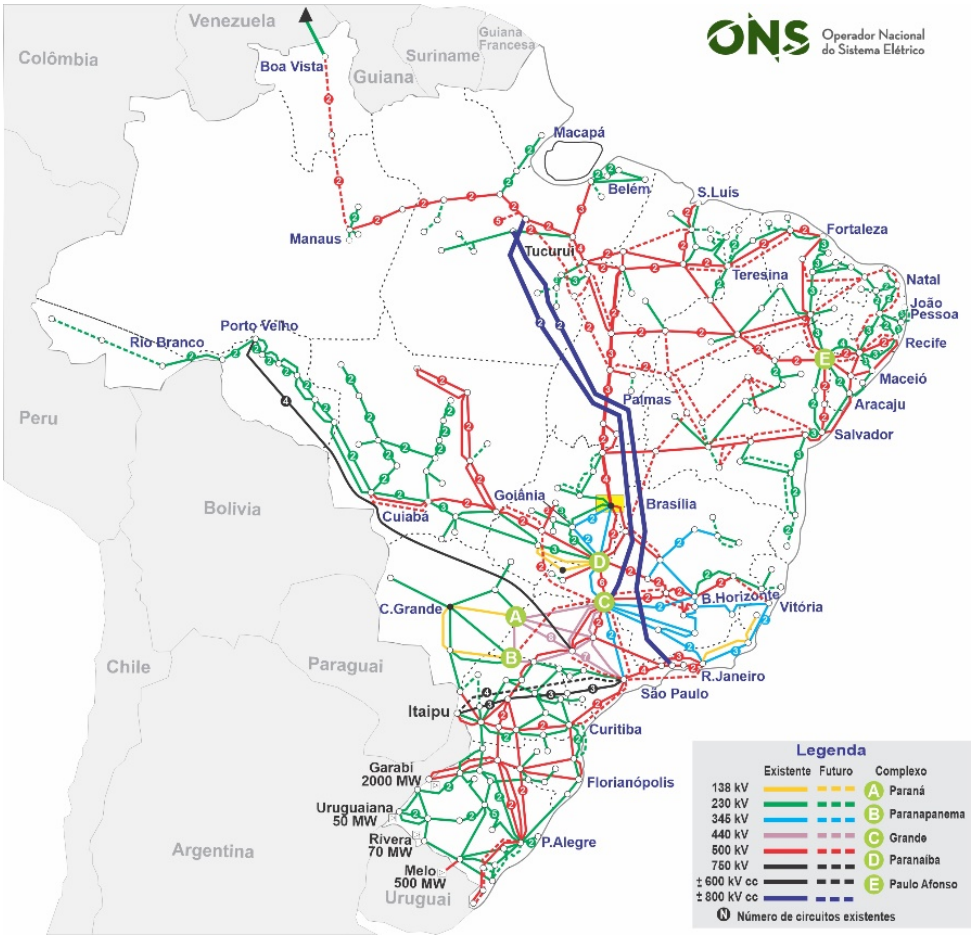
MW	Megawatt
NAN	*Not a Number*
NTPS-II	Usina Termelétrica de Neyveli, unidade 2
ONS	Operador Nacional do Sistema Elétrico
PACF	Função de Autocorrelação Parcial
R2	Coeficiente de Determinação
RNA	Redes Neurais Artificiais
RNN	Redes Neurais Recorrentes
SAGE	Sistema Aberto de Gerenciamento de Energia
SE	Subestação de Energia
SEP	Sistemas Especiais de Proteção
SIN	Sistema Interligado Nacional
STLF	*Short Term Load Forecaster* ou Previsor de carga de curto-prazo
SVM	*Support Vector Machine*
SVM	*Support Vector Regression*
XGBoost	*eXtreme Gradient Boosting*

SUMÁRIO

1 INTRODUÇÃO

O Sistema Interligado Nacional (SIN) é um sistema elétrico de potência extremamente complexo: Envolve desde a Geração de grandes blocos de energia nas unidades geradoras das usinas, toda a rede de Transmissão de Energia (também chamada de Rede Básica[1]), composta pelas linhas de transmissão em alta tensão, e até os consumidores industriais e residenciais na rede de distribuição. Portanto, é fundamental que o planejamento da operação desse sistema seja feito da maneira mais precisa possível, minimizando as perdas, otimizando a utilização dos recursos sistêmicos disponíveis, reduzindo as falhas e, para o Agente de Transmissão, maximizando os resultados coorporativos necessários para a sustentabilidade do negócio, como é o caso da Eletrobras Eletronorte. A Figura 1 apresenta o sistema de transmissão do SIN.

Figura 1 – O Sistema Interligado Nacional - Horizonte de Expansão 2024.



Fonte: Operador Nacional do Sistema Elétrico - ONS.

[1] Rede Básica é composta pelas instalações de transmissão do SIN, sob concessão das transmissoras, definida segundo critérios estabelecidos pela ANEEL (ANEEL, 2022).

O Operador Nacional do Sistema Elétrico (ONS) é o órgão responsável pelo planejamento, coordenação e controle da operação das instalações de geração e transmissão de energia elétrica no SIN (ONS, 2021). Entre as suas responsabilidades estão a elaboração dos Estudos de Planejamento Elétrico do SIN de curto prazo (com horizontes de planejamento mensais e quadrimestrais) e de médio prazo (com horizonte quinquenal). Nestes relatórios são apresentados os resultados das avaliações do desempenho elétrico do SIN, tanto em condições normais como em situações de contingências, tendo como base os programas de obras de transmissão e geração, o cronograma de manutenção dos equipamentos de transmissão informados pelos Agentes de Transmissão, bem como as previsões de carga informadas pelas empresas de Distribuição, conforme ANEEL (2016) e consolidadas pelo ONS, segundo os requisitos de ONS (2022b). Como produto destes estudos, destacam-se as medidas operativas e as instruções de operações, que são as diretrizes normativas utilizadas para a coordenação da operação do SIN e utilizados pelos Centros de Operação do ONS e dos Agentes de Transmissão e Geração distribuídos em todo território nacional; a indicação da necessidade de reforço no sistema de transmissão; a indicação da necessidade de revisão ou concepção de Sistemas Especiais de Proteção (SEP) ou dos ajustes de proteções convencionais e de dispositivos de controles, a fim de garantir a operação segura e otimizada do SIN (ONS, 2022a); e finalmente, disponibilizar os casos base de simulação em regime permanente, utilizados nos estudos de planejamento, com os diversos patamares de carga previstos em cada horizonte avaliado, para uso geral, inclusive dos Agentes de Transmissão (ONS, 2020b).

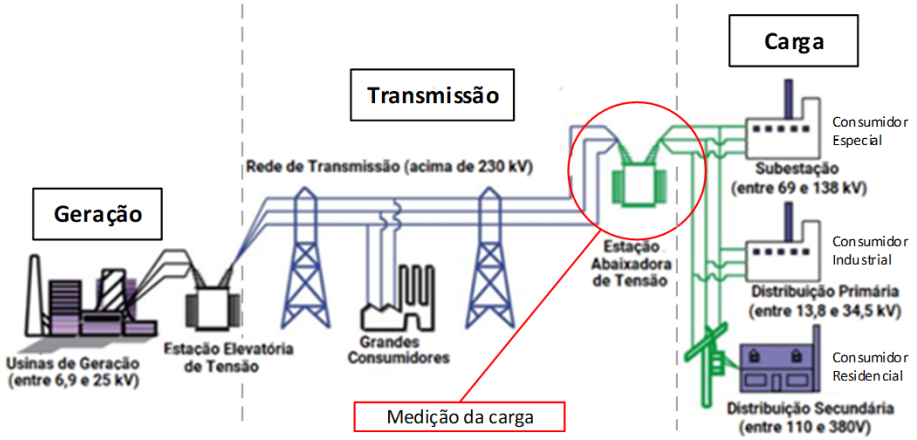
O Agente de Transmissão (ou Transmissora de Energia Elétrica, ou Concessionária de Transmissão), pessoa jurídica titular de concessão para a prestação do serviço público de transmissão de energia elétrica (ONS, 2020a), realiza o planejamento da operação de seus ativos buscando aliar o atendimento aos critérios e requisitos definidos nos Procedimentos de Rede[2] do ONS – com o cronograma de manutenções dos seus ativos de transmissão, quais sejam, linhas de transmissão, transformadores de potência, disjuntores e chaves seccionadoras, reatores e bancos de capacitores, módulos gerais de subestações e etc. Portanto, é fundamental realização de uma programação eficiente das intervenções, isto é, da parada dos ativos para manutenção, garantindo a confiabilidade do sistema, a vida útil dos equipamentos de transmissão e a continuidade do atendimento ao consumidor.

Para isto, é fundamental que se conheça bem o comportamento da demanda de energia do consumidor – em outras palavras, a carga elétrica – dada pelo fluxo de potência nos transformadores de fronteira, ou seja, dos transformadores que entregam energia elétrica

[2] Os Procedimentos de Rede são as regras propostas pelo ONS, com participação dos agentes e aprovação da ANEEL, para coordenação e controle da operação da geração e da transmissão de energia elétrica integrantes do SIN, de acordo com a Lei nº 9.648, de 17 de maio de 1998. Estabelece os procedimentos e os requisitos técnicos necessários para o planejamento, implantação, uso e operação do SIN, bem como as responsabilidades do ONS e dos agentes.

para a rede de distribuição, conforme ilustra a Figura 2. Uma programação ineficiente da Transmissora leva o ONS a indeferir a autorização do desligamento do equipamento a ser manutenido, o que traz um risco para o equipamento e, portanto, para o negócio e, em último caso, para o sistema, no caso extremo de um sinistro pela falta de manutenção.

Figura 2 – Simplificação do sistema elétrico de potência.



Fonte: Adaptado de Kagan, Oliveira e Robba (2021).

1.1 Justificativa e Motivação

Para a programação de uma intervenção, atualmente, a Eletrobras Eletronorte adota uma metodologia baseada na previsão de carga dos casos base do ONS. Entretanto, por mais que a consolidação da previsão de carga para os horizontes dos estudos de planejamento do ONS seja robusta o suficiente para realizar a avaliação do desempenho do sistema nos patamares mais relevantes dentro do período analisado (carga pesada, média e leve, com variações dentro do horizonte para o período de máximo e mínimo despacho nas usinas por questões hidrológicas), esta previsão é incapaz de descrever a variação temporal discreta da carga ao longo do tempo.

Além disso, estes casos não consideram as condições de carga não-coincidentes. Isto significa que o caso de carga mais pesada não traduz a demanda máxima solicitada por um dado consumidor de fronteira, mas a maior demanda total para a ponta de carga daquela região, o que pode prejudicar a análise da programação da parada de um equipamento de uma instalação específica, cuja demanda máxima não seja coincide com a ponta de carga da região.

Por isso, adicionalmente, a Eletronorte tem complementado as análises considerando o registro histórico da carga para o período equivalente à janela programada. Isto é, além das análises por meio da simulação em regime permanente de contingência simples ou duplas dos equipamentos a serem desligados por meio dos casos base do ONS, é realizada uma análise complementar considerando os dados históricos de medição para a semana

anterior à programação, ou até mesmo para a semana equivalente no mês para o ano anterior, considerando toda a variação temporal da carga *versus* a capacidade remanescente da instalação após o desligamento do ativo, a fim de verificar se é possível acomodar o desligamento naquele período sem prejuízo para o sistema e para o consumidor.

Essa análise complementar melhora a sensibilidade da análise do planejamento da intervenção, mas ainda não é suficiente. A utilização de dados históricos, sem realizar qualquer tipo de análise preditiva, esperando que o comportamento da carga seja similar, leva a algumas imprecisões indesejadas que, se não levam ao indeferimento da solicitação de intervenção, pode levar a interrupção do serviço por necessidade sistêmica, quando, por exemplo, a carga cresce acima do esperado durante o desligamento do equipamento, levando os demais à sobrecarga.

Diante deste problema, evidencia-se a necessidade da utilização de técnicas capazes de prever com precisão o comportamento da carga futura, baseando-se em dados históricos de medição. Dada a importância desta questão, não apenas para o problema do planejamento do Agente de Transmissão em realizar a manutenção de seus ativos, como também para todo o Setor Elétrico que depende da previsibilidade da carga para operar, planejar a expansão, otimizar o preço e negociar a energia e etc., bem como o avanço da capacidade computacional de processamento e dos algoritmos de *machine learning*, que diversas técnicas tem sido desenvolvidas, no sentido de encontrar uma solução para a previsão de carga no longo-prazo (anos à frente), no médio-prazo (meses à frente), no curto-prazo (semana ou dias a frente) e curtíssimo-prazo (horas ou minutos a frente). Este trabalho, portanto, visa contribuir com esta discussão.

1.2 Questão de Pesquisa e Objetivos

O objetivo deste trabalho é apresentar a melhor metodologia para realizar predição de carga elétrica em sistemas elétricos de potência, considerando os registros históricos de demanda em MW disponíveis. Dessarte, formulou-se a seguinte questão de pesquisa, para fim de norteamento do trabalho:

Qual é a melhor estratégia para realizar a predição de carga elétrica com boa precisão, por meio dos dados históricos de medição instantânea diária dos meses pregressos, utilizando ferramentas de inteligência artificial/machine learning?

Visando de desenvolver um trabalho capaz de responder tal pergunta, foram definidos alguns objetivos específicos, quais sejam:

- Realizar revisão bibliográfica a fim de mapear algumas das principais técnicas de previsão de carga em sistemas elétricos de potência;
- Definir o estudo de caso de previsão para um determinado barramento atendido por

subestação da Eletrobras Eletronorte, realizar o tratamento dos dados históricos de medição e submeter os dados à cada algoritmo de previsão anteriormente mapeado, considerando diversos horizontes, ou seja, curto, médio e longo-prazo;

- Avaliar os resultados e validá-los por meio da comparação das saídas dos modelos preditivos com a carga realizada para seu respectivo horizonte.

1.3 Organização do Trabalho

Este trabalho está estruturado em 6 capítulos, conforme descrição a seguir:

O Capitulo 1 contextualiza e apresenta a motivação do trabalho: A importância de realizar a previsão de carga para o sucesso do planejamento da operação de um sistema elétrico de potência. É apresentado também a Questão de Pesquisa, bem como as etapas necessárias para alcançar o objetivo do trabalho.

O Capitulo 2 apresenta os conceitos fundamentais dos principais elementos abordados neste trabalho, que são as Cargas Elétricas, isto é, a quantidade de potência ativa entregue à concessionária de energia num dado instante de tempo, em MW; e os principais modelos preditores de carga, segundo a literatura.

O Capítulo 3 apresenta o Estado-da-Arte dos preditores de carga. São apresentados os estudos mais recentes de preditores de curto, médio e longo prazo.

O Capitulo 4 aborda toda a metodologia utilizada em cada etapa percorrida na construção do conhecimento - da aquisição dos dados de medição histórica de carga, o tratamento desses dados, a adição de novas características, toda a fundamentação das análises exploratórias visando extrair os padrões dos dados, a modelagem de cada preditor estudado e, finalmente, os critérios de avaliação dos resultados.

O Capítulo 5 apresenta os resultados dos testes analíticos de exploração dos dados, propostos no capítulo anterior. Apresenta também o resultado da aplicação de cada modelo sobre o conjunto de dados. Cada resultado é avaliado sob os critérios definidos no capítulo de Metodologia. Finalmente, é proposta uma discussão sobre a influência objetiva de algumas características do conjunto de dados na qualidade da previsão da carga.

O Capitulo 6 apresenta as conclusões deste trabalho, buscando responder a questão de pesquisa proposta anteriormente. Ao final, é apresentado uma proposta de trabalho futuro.

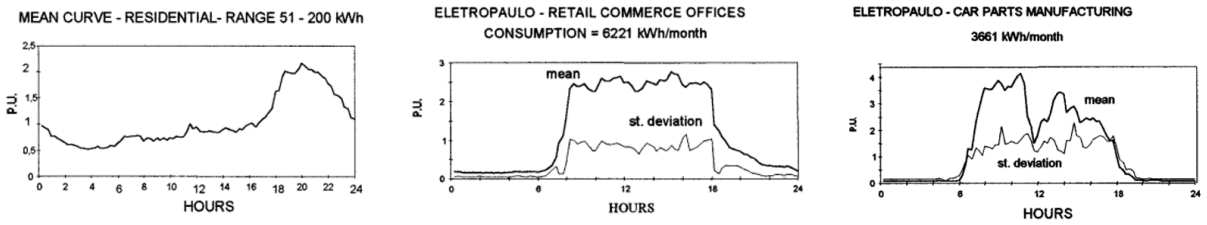
2 FUNDAMENTAÇÃO TEÓRICA E TRABALHOS RELACIONADOS

2.1 A carga elétrica como variável no tempo

A carga atendida por uma subestação é uma grandeza não-linear. Varia no tempo conforme o perfil de consumo daquela determinada região, sendo sensível a fatores econômicos, climáticos e até mesmo culturais.

Na distribuição é bem característica a variação da carga ao longo do dia dependendo do tipo do consumidor. Na Figura 3, vemos o perfil de carga de consumidores residenciais, comerciais e industriais:

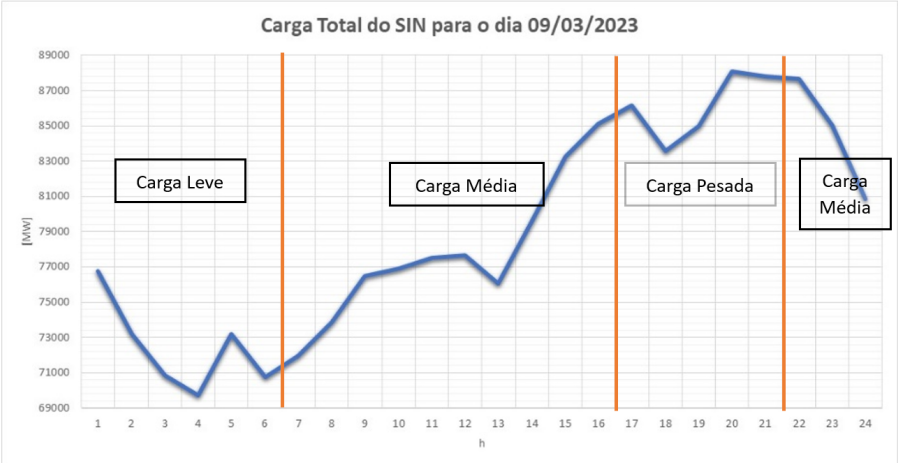
Figura 3 – Características temporais da carga baseada no tipo de consumidor.



Fonte: Jardini *et al.* (2000).

Na Rede Básica, por outro lado, por se tratar de um ponto de medição à montante da distribuição, o comportamento da carga verificado não necessariamente tem essa característica tão bem definida, apesar de se perceber variações ao longo do tempo. A Figura 4 ilustra a variação diária da carga total do sistema para um dia útil.

Figura 4 – Variação da carga no SIN para o dia 09/03/2023.



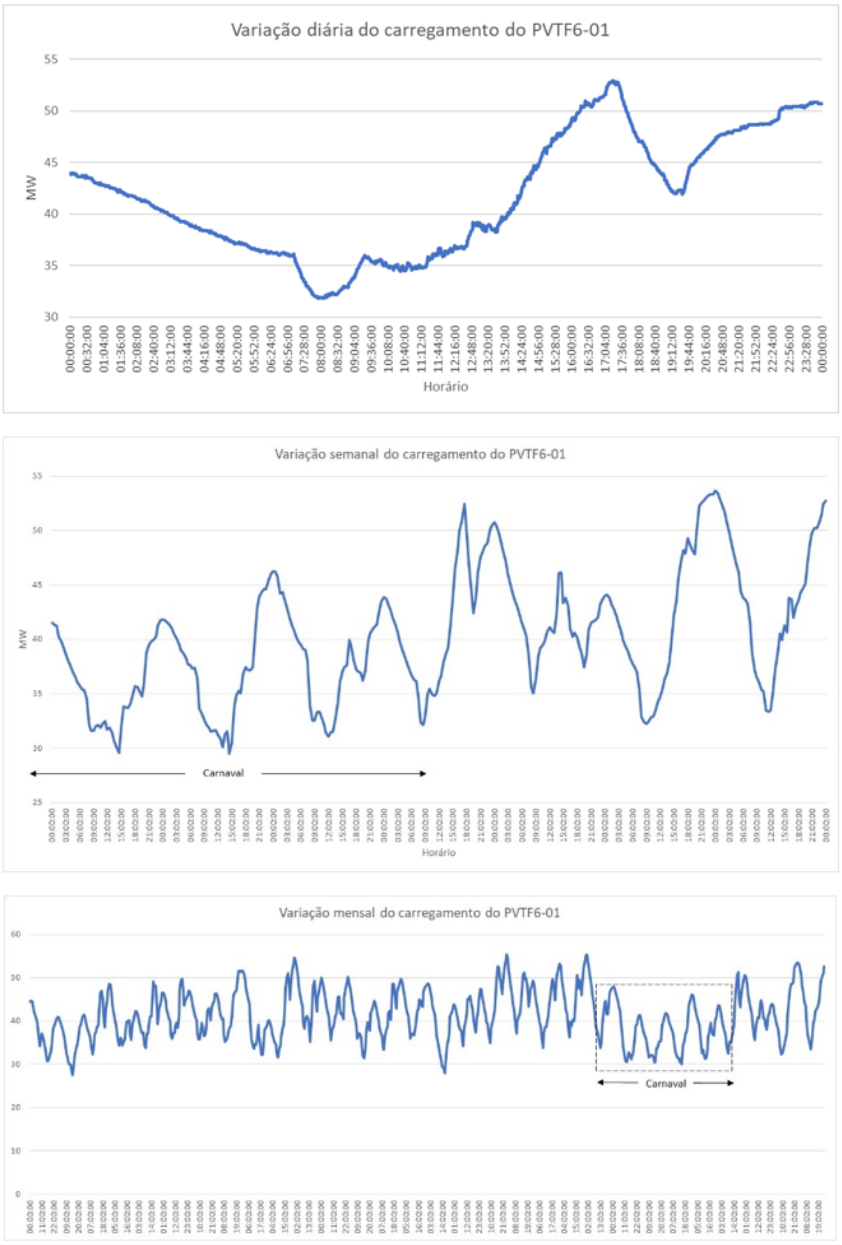
Fonte: ONS em ONS (2023).

O conceito de carga leve, média e pesada é ligada ao perfil de consumo de energia do sistema e está intrinsecamente relacionado à faixa horária do dia. Em ONS (2020a)

são definidos os patamares de carga em relação ao horário do dia, conforme ilustrado na Figura 4. Já o conceito de ponta de carga está relacionado à maior carga medida naquele dia, que na mesma figura é de 88 GW, aproximadamente.

A Figura 5 apresenta um exemplo de variação temporal da carga, verificado através da variação do carregamento de um dos transformadores da SE Porto Velho, monitorado por um dia, uma semana e um mês.

Figura 5 – Variação temporal do carregamento de um transformador de fronteira.



Fonte: do próprio autor. Dados cedidos pela Eletrobras Eletronorte.

Da Figura 5, verificamos que a carga ao longo do tempo apresenta uma tendência e varia de forma periódica, e que fatores como os finais de semana e feriados – uma vez que alteram o perfil de consumo da energia naquele dia específico – influenciam seu

comportamento global.

Visto que o perfil de carga nestes períodos de finais de semana e feriados é mais baixo, muitas vezes as programações de parada de equipamentos para manutenção devem ser planejadas para estes períodos. Por isso, é fundamental que qualquer algoritmo de previsão de carga seja capaz de reproduzir este comportamento. No tópico a seguir serão apresentadas as metodologias mais consagradas de previsão de carga na literatura.

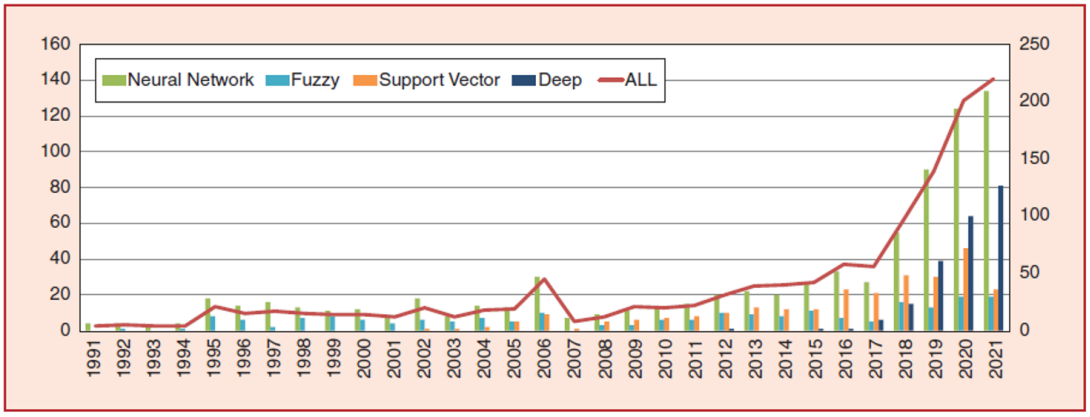
2.2 Técnicas e Metodologias utilizadas na previsão de carga

O problema de previsão de carga não é novo e existe literatura a respeito do assunto desde os primórdios da eletrificação urbana em grande escala, nas primeiras décadas do século XX. Durante primeira metade do século passado, eram utilizados diversos métodos que iam desde a aplicação de percentuais anuais constantes de crescimento nos registros históricos para previsões de longo prazo (RALSTON, 1925), até a modelagem dinâmica da carga para estudos de previsão no curto-prazo.

Com a evolução da capacidade computacional a partir dos anos 1960, os métodos estatísticos passaram a ser mais utilizados, tanto para realizar o ajuste do modelo dinâmico da carga ao longo do tempo, como para realizar a previsão em si por meio das medidas históricas de carga (HONG; WANG, 2022). Essas técnicas ainda são relevantes e trabalhos importantes utilizando essa metodologia para previsão de carga ainda são publicados.

No final dos anos 1980 e começo dos anos 90, os primeiros trabalhos de previsão de carga utilizando métodos de Inteligência Artificial (IA) e Aprendizado de Máquina (ou *Machine Learning* – ML) foram publicados, baseados em técnicas como Lógica *Fuzzy*, *Support Vector Machine* (SVM) e Redes Neurais Artificiais (RNA). Na Figura 6 é mostrada a evolução da publicação dos trabalhos de previsão de carga por meio de técnicas de IA e ML ao longo dos anos.

Figura 6 – Trabalhos sobre previsão de carga utilizando técnicas de IA ou *machine learning* até 2021.



Fonte: Hong e Wang (2022).

Sendo assim, em virtude do avanço do estudo das técnicas de inteligência artificial e aprendizado de máquina, viabilizadas pelo desenvolvimento das ferramentas computacionais que permitiram não só a implementação de metodologias cada vez mais complexas, como também a manipulação de dados com volumes cada vez maiores, nunca foi tão profícuo o estudo de previsores de carga como tem sido hoje em dia. Contudo, antes de ser apresentado o estado-da-arte deste segmento, serão apresentados os fundamentos teóricos de algumas das técnicas mais comuns verificadas na literatura.

De acordo com Singh *et al.* (2012), as técnicas de previsão de carga podem ser agrupadas em três grupos principais: Técnicas Tradicionais de Previsão, Técnicas Tradicionais Modificadas e Técnicas de *Soft Computing*. Essa divisão será respeitada na apresentação dos conceitos das principais metodologias adotadas:

2.2.1 Técnicas Tradicionais de Previsão

Conforme Singh *et al.* (2013), figuram entre as principais técnicas tradicionais de previsão de carga os métodos estatísticos de Regressão (Linear e Múltipla), por meio do emprego de modelos de correlação entre a carga e outros fatores que influenciam a sua variação. O princípio fundamental da Regressão é correlacionar linearmente dados de entrada com os dados de saída. É, basicamente, estimar os coeficientes de uma equação linear que melhor descreve (ou melhor prediz) os valores de saída (FERRAZ, 2022).

2.2.1.1 Método de Regressão

Os Métodos de Regressão são umas das técnicas mais amplamente utilizadas entre as técnicas estatísticas pela simplicidade em sua implementação. Esta metodologia é normalmente empregada para modelar a relação entre a carga e um outro fator, como condições climáticas, tipo de carga, tipo do dia etc. Sua expressão matemática pode ser escrita como:

$$L(t) = Ln(t) + \sum a_i x_i(t) + e_t, \tag{2.1}$$

em que Ln(t) é carga normalizada no instante $'t'$, a_i é o coeficiente variável estimado ou parâmetro de regressão, $x_i(t)$ é fator independente influenciador ou variável preditora e e_t é o componente de ruído branco ou erro aleatório. Entretanto, de acordo com Silva (2022), existe uma desvantagem na utilização desta técnica em virtude "da não linearidade e complexidade da relação entre a carga de demanda e as variáveis influenciadoras. Além disso, para conseguir resultados razoavelmente precisos é necessário um poder computacional elevado".

Conforme Singh *et al.* (2012), a precisão do método depende da adequada representação das possíveis condições futuras por meio dos dados históricos, embora seja simples

criar uma medida para detectar quaisquer previsões incorretas. O processo requer poucos parâmetros, que são facilmente calculáveis a partir dos dados históricos usando a técnica de validação cruzada (*cross validation*). Para prever a carga com precisão ao longo de um ano, deve-se levar em conta a variação sazonal, o crescimento anual e a variação diária mais recente da carga.

2.2.1.2 Regressão Múltipla

A Regressão múltipla é outra técnica bastante difundida e utilizada para realizar a predição da carga quando esta é afetada por vários fatores externos concomitantemente (fatores climáticos, econômicos, etc). A representação matricial do método é dada por:

$$Y = X\beta + \epsilon, \tag{2.2}$$

onde, Y é vetor das variáveis dependentes ou das variáveis resposta, X é a matriz das variáveis independentes ou das variáveis preditoras, b é o vetor dos parâmetros de regressão, e, ϵ é o vetor dos erros.

Entre alguns trabalhos que utilizaram desta metodologia para realizar a previsão de carga, destaca-se Papalexopoulos e Hesterberg (1990), onde é apresentado um modelo de regressão múltipla capaz de determinar a ponta de carga como uma combinação linear das cargas máximas do primeiro, do segundo, do sexto e do sétimo dias anteriores, por meio da determinação dos coeficientes através da estimativa de mínimos quadrados.

2.2.2 Técnicas Tradicionais Modificadas

Conforme Singh *et al.* (2012), as técnicas tradicionais de previsão foram aperfeiçoadas, sendo capazes de corrigir automaticamente os parâmetros do modelo preditor sob variação das condições de contorno. Algumas dessas principais técnicas são as Séries Temporais Estocásticas e a *Support Vector Machine* (SVM).

2.2.2.1 Séries Temporais Estocásticas ou Modelos Estocásticos

De acordo com de COSTA (2017), uma série temporal é uma sequência de observação organizados sequencialmente no tempo, como é o registro histórico de medição de carga ao longo de um período definido, por exemplo. Entre os preditores de carga de curto prazo (SINGH *et al.*, 2012) e de longo prazo (COSTA, 2017), a utilização de séries temporais estocásticas é um dos métodos mais difundidos. Essa metodologia baseia-se na premissa de que os registros históricos possuem uma estrutura interna, como autocorrelação, tendência e variação sazonal. Desse modo, busca-se montar com precisão um padrão que corresponda aos dados disponíveis e, em seguida, usar o modelo formado para obter o valor previsto em relação ao tempo (SINGH *et al.*, 2012). A seguir, são apresentados alguns dos métodos de previsão por meio de séries temporais:

2.2.2.1.1 Modelo Autoregressivo (*Autoregressive Model* - AR):

O modelo Autoregressivo é capaz representar o perfil de carga, assumindo que a variável predita é uma combinação linear da medida dos instantes anteriores. Seu equacionamento, apresentado em Box *et al.* (2015), é dado por:

$$\tilde{z}_t = \phi_1 \tilde{z}_{t-1} + \phi_2 \tilde{z}_{t-2} + \cdots + \phi_p \tilde{z}_{t-p} + a_t, \tag{2.3}$$

sendo:

$\tilde{z}_t$: carga prevista no instante t;

$\phi_1, \ldots, \phi_p$: coeficientes autorregressivos;

p: ordem ou horizonte do modelo autorregressivo;

$a_{t:}$: ruído branco.

2.2.2.1.2 Modelo de Média Móvel (*Moving Average Models* - MA):

Ainda de acordo com a definição proposta por Box *et al.* (2015), o modelo autorregressivo expressa o desvio $\tilde{z}_t$ do processo como uma soma ponderada finita de p defasagens anteriores $\tilde{z}_{t-1}$, $\tilde{z}_{t-2}$, ..., $\tilde{z}_{t-p}$ do processo, somado a um erro aleatório $a_{t:}$. De forma análoga, $\tilde{z}_t$ é descrito como uma soma ponderada infinita dos a's.

Outro tipo de modelo, de grande importância prática na representação de séries temporais observadas, é o processo de média móvel finita. Aqui, $\tilde{z}_t$ é linearmente dependente de um número finito q de a's anteriores e é chamado de processo de média móvel de ordem q:

$$\tilde{z}_t = a_t - \theta_1 a_{t-1} - \theta_2 a_{t-2} - \cdots - \theta_q a_{t-q}, \tag{2.4}$$

em que $\theta_1, \theta_2, \ldots, \theta_q$ são os coeficientes da média móvel.

2.2.2.1.3 Modelo Autoregressivo de Média Móvel (*Autoregressive Moving Average Model* - ARMA):

No modelo ARMA foi concebido para alcançar maior flexibilidade na adaptação das series temporais, combinando os termos autorregressivos quanto de média móvel no modelo (BOX *et al.*, 2015). Assim, temos o seguinte equacionamento:

$$\tilde{z}_t = \phi_1 \tilde{z}_{t-1} + \cdots + \phi_p \tilde{z}_{t-p} + a_t - \theta_1 a_{t-1} - \cdots - \theta_q a_{t-q}. \tag{2.5}$$

2.2.2.1.4 Modelo Autoregressivo Integrado de Média Móvel (*Autoregressive Integrated Moving Average Model* - ARIMA)):

Os modelos AR, MA e ARMA são processos conhecidos como processos estacionários, isto é, quando a média e a variância da série temporal se mantêm constante ao longo da janela de tempo. Para séries com tendência e sazonalidade, ou seja, não estacionárias, foi desenvolvido o modelo ARIMA.

Para isso, o modelo realiza uma transformação da série para a forma estacionária por meio da operação:

$$w_t = \nabla^d z_t, \tag{2.6}$$

em que d varia discretamente entre 0 e 2 para representar a não-estacionariedade ($d = 0$ significa que a série é estacionária).

Assim, o modelo pode ser finalmente representado seguindo a equação:

$$w_t = \phi_1 w_{t-1} + \cdots + \phi_p w_{t-p} + a_t - \theta_1 a_{t-1} - \cdots - \theta_q a_{t-q}. \tag{2.7}$$

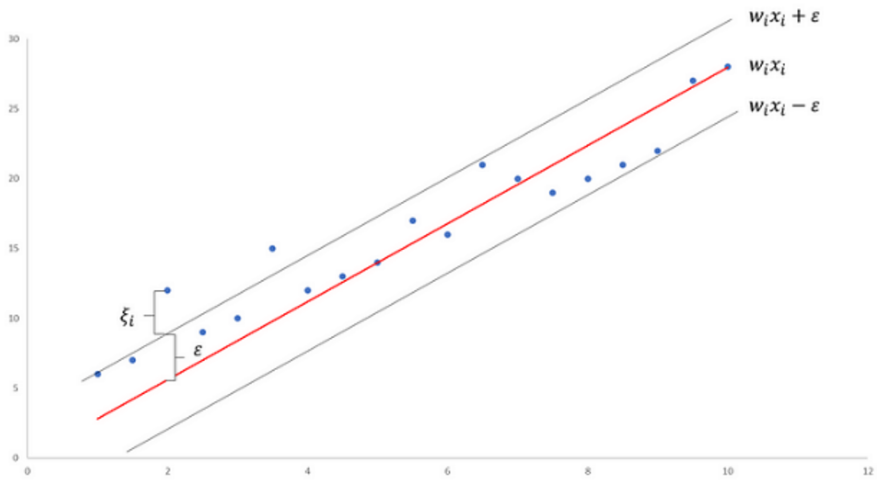
Todos os modelos de séries temporais estocásticas apresentados anteriormente dependem apenas dos dados históricos de medição da carga para realizar a predição. No entanto, para quando há fatores externos correlacionados (ou dados exógenos) aos dados principais, foram desenvolvidos aperfeiçoamentos do ARMA e ARIMA, que são o ARMAX e o ARIMAX, respectivamente.

2.2.2.2 Técnicas baseadas em *Support Vector Machine* – SVM

Este método foi apresentado em Cortes e Vapnik (1995) e foi concebido para resolver problemas de classificação. Posteriormente, sua formulação foi estendida para também resolver problemas de regressão (DRUCKER *et al.*, 1996; VAPNIK; GOLOWICH; SMOLA, 1996). Por isso é comum encontrarmos na literatura o termo SVR, de *Support Vector Regression*. Sua saída é contínua e numérica e, portanto, encontra aplicação no problema de previsão de carga. O objetivo da ferramenta é encontrar a melhor curva que se ajusta aos dados, de modo a prever os valores futuros.

O SVR baseia-se na ideia de que o melhor ajuste para os dados é aquele que maximiza a distância entre as amostras mais próximas da curva e a própria curva. Essas amostras são chamadas de vetores de suporte. A curva é encontrada através da otimização de uma função objetivo que minimiza o erro de previsão por meio de uma função de custo e maximiza a margem. Para realizá-la o modelo utiliza um conjunto de parâmetros que controlam a complexidade da curva e a largura da margem.

Figura 7 – Ajuste dos dados pela SVR.



Fonte: Sharp (2023).

A equação do SVR é dada por (CEPERIC; CEPERIC; BARIC, 2013):

$$f(x) = \sum_{i=1}^{l} w_i K(x_i, x) + b, \tag{2.8}$$

onde:

W_i : coeficiente de Lagrange associado ao vetor de suporte;

X_i : vetor de entrada p-dimensional associado à saída;

K: função kernel (Linear, polinomial ou RBF (Radial Basis Function, Sigmoidal);

l: número dos dados de amostra de treinamento.

Conforme Domingos *et al.* (2019), a utilização de funções kernel K possibilita o mapeamento não-linear dos dados para um espaço de características no qual uma função linear é encontrada. Na Tabela 1 são apresentadas as funções Kernel mais utilizadas:

Tabela 1 – Funções Kernel mais utilizadas.

Tipo de Kernel	Função $K(x_i, x)$	Parâmetros
Polinomial	$(\delta(x_i \cdot x) + \kappa)^d$	$\delta, \kappa ed$
Gaussiano	$\exp(-\sigma\|\|x_i - x\|\|^2)$	σ
Sigmoidal	$\tanh(\delta(x_i \cdot x) + \kappa)$	$\delta e \kappa$

Fonte: Lorena e Carvalho (2007).

Entre alguns trabalhos que utilizaram a ferramenta SVR para realizar a previsão de carga estão o de Barros (2014), que comparou as técnicas de modelos autorregressivos, redes neurais e o SVR, obtendo os melhores resultados por meio desta última técnica para

os dados propostos; e Chen, Chang *et al.* (2004), que venceu a competição da EUNITE [1] em 2001, ao obter a melhor predição de uma carga de médio prazo (carga máxima diária de 31 dias à frente), através de um modelo SVM. O trabalho de Domingos *et al.* (2019) realiza predições de carga no contexto do sistema elétrico brasileiro em três horizontes diferentes, utilizando um modelo conjunto de um *Perceptron* de Múltiplas Camadas (MLP) e um SVM, obtendo bons resultados.

2.2.3 Técnicas de *Soft Computing* e Aprendizado de Máquina (*Machine Learning*)

Técnicas de *Soft Computing* são um conjunto de métodos e algoritmos computacionais que se baseiam em abordagens heurísticas e aproximativas para resolver problemas complexos, muitas vezes caracterizados por dados incertos, incompletos, ambíguos ou imprecisos. Diferentemente dos métodos tradicionais de computação, que se baseiam em lógica matemática exata e algoritmos determinísticos, as técnicas de *Soft Computing* são voltadas para a simulação de processos cognitivos humanos, como a capacidade de raciocinar sob incertezas, aprender a partir de exemplos e adaptar-se a novas situações.

O Aprendizado de Máquina, ou *Machine Learning*, é um campo de estudo que explora o desenvolvimento de algoritmos que melhoram automaticamente através da experiência. Isso é conseguido ao treinar o modelo, expondo-o a dados rotulados, e permitindo que o modelo aprenda e faça inferências a partir desses dados.

Dentro do Aprendizado de Máquina, um subcampo particularmente relevante é o de métodos de aprendizado em conjunto. Esses métodos operam construindo uma sequência de modelos, tipicamente árvores de decisão, e os combinando de tal forma que os pontos fracos de um modelo são corrigidos pelos modelos subsequentes. A Árvore de Decisão é uma abordagem de aprendizado de máquina que se baseia em uma estrutura de árvore, onde cada nó interno representa um "teste"em um atributo, cada ramo representa o resultado desse teste e cada folha representa uma previsão de classe.

Dois exemplos de técnicas de aprendizado em conjunto que se baseiam em árvores de decisão são a Floresta Aleatória ou *Random Forest* e a XGBoost. Ambas são métodos poderosos e flexíveis que podem capturar padrões complexos nos dados, e são amplamente utilizadas em uma variedade de problemas de classificação e previsão.

No âmbito do *Soft Computing* e Aprendizado de Máquina, existem várias técnicas que incluem Redes Neurais Artificiais, Lógica *Fuzzy*, Algoritmos Genéticos, Sistemas Imunológicos Artificiais e Sistemas Especialistas.

Redes Neurais Artificiais, por exemplo, são sistemas computacionais inspirados no funcionamento do cérebro humano, capazes de aprender a partir de exemplos e reconhecer padrões em dados complexos. Elas representam um ponto de convergência entre o

[1] European Network on Intelligent Technologies for Smart Adaptive Systems

Aprendizado de Máquina e *Soft Computing*, pois tanto têm a capacidade de aprender a partir de dados, como se baseiam na simulação de processos cognitivos humanos. A Lógica *Fuzzy* é uma técnica que permite lidar com informações imprecisas e ambíguas, por meio de uma lógica baseada em graus de pertinência. Algoritmos Genéticos são técnicas de otimização inspiradas no processo de evolução biológica, que buscam soluções melhores a partir da seleção natural e da combinação de características de soluções candidatas. Sistemas Imunológicos Artificiais são inspirados no sistema imunológico biológico, e buscam detectar e combater ameaças, como invasores ou anomalias, em sistemas computacionais. Sistemas Baseados em Conhecimento são sistemas que utilizam um conjunto de regras e conhecimentos específicos para tomar decisões em situações complexas.

A seguir serão apresentadas algumas das técnicas mais relevantes na literatura aplicadas ao problema de previsão de carga elétrica.

2.2.3.1 Lógica *Fuzzy*

A Lógica *Fuzzy* é uma abordagem de lógica matemática que permite representar incerteza e imprecisão usando valores contínuos que variam de 0 a 1, em oposição à lógica Booleana clássica, que permite apenas valores binários de 0 ou 1. Na tentativa de modelar o raciocínio humano, a Lógica *Fuzzy* permite a atribuição de variáveis linguísticas (COSTA, 2017).

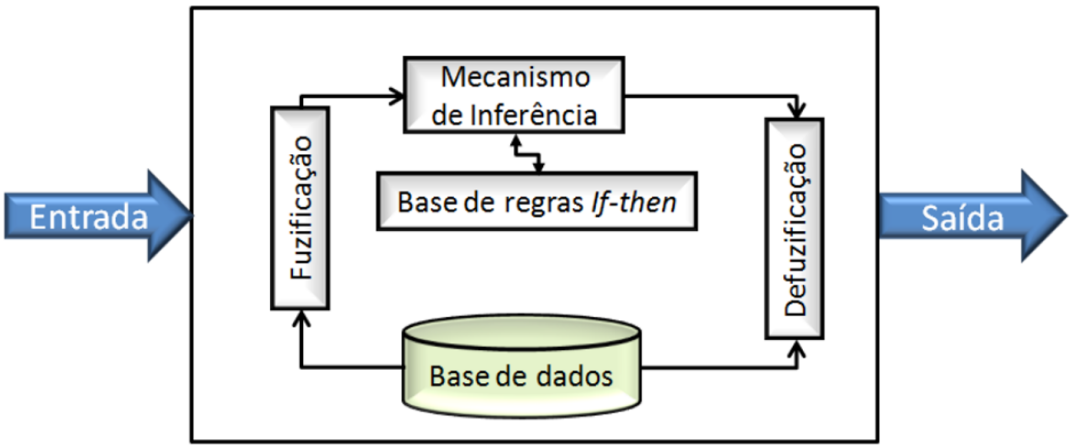
Na Lógica *Fuzzy*, um valor de pertinência é atribuído a cada elemento de um conjunto para significar o grau em que ele pertence a esse conjunto. As funções de pertinência, que são curvas matemáticas que definem a relação entre um valor de entrada e seu grau de pertinência, podem ser usadas para determinar esses valores de pertinência. Os trabalhos de Mamdani e Assilian (1975), Takagi e Sugeno (1985) e Chien (1990) apresentam um extenso arcabouço conceitual desta metodologia. A seguir, uma breve fundamentação:

A Lógica *Fuzzy*, portanto, consiste em três componentes principais: o conjunto *Fuzzy*, a função de pertinência e as operações *Fuzzy*. Um conjunto *Fuzzy* é uma coleção de objetos com vários graus de pertencimento a este conjunto. Cada elemento em um conjunto *Fuzzy* tem um grau de pertinência entre 0 e 1 que indica o quão bem ele corresponde à descrição do conjunto. A função de pertinência é quem determina este atributo de pertinência de um elemento em um conjunto *Fuzzy*, por meio da aplicação de uma função triangular, função trapezoidal ou função gaussiana. Por fim, as operações *Fuzzy* são usadas para criar novos conjuntos *Fuzzy* que podem ser aplicados para modelar e analisar sistemas complexos. União, interseção e complemento *Fuzzy* são operações amplamente difundidos.

Em Mamdani e Assilian (1975) é proposta a estrutura do sistema de inferência, que é composto por cinco componentes principais: a interface de fuzificação, a base de regras, o mecanismo de inferência, o banco de dados e a interface de defuzificação, como

ilustra a Figura 8.

Figura 8 – Estrutura do sistema de inferência da Lógica *Fuzzy*.



Fonte: Zimmermann (2011).

A fuzzificação é usada para transformar valores numéricos em conjuntos *Fuzzy*, de modo que variáveis contínuas e incertas possam ser incorporadas em modelos de Lógica *Fuzzy*. Para cada valor de entrada, o procedimento compreende atribuir graus de pertinência a um conjunto *Fuzzy* através da função de pertinência. Cada tipo de função de pertinência tem suas próprias propriedades, que são selecionadas com base no domínio do problema e nos requisitos do modelo.

No mecanismo de inferência, as regras *Fuzzy* são utilizadas para inferir a saída do modelo *Fuzzy*. Normalmente, as regras *Fuzzy* são definidas usando uma base de regra *if-then*, em que as cláusulas *if* refletem os critérios que devem ser atendidos para que a regra seja aplicada e a cláusula *then* define o resultado da regra.

Essas regras podem ser expressas em termos de conjuntos *Fuzzy*, usando funções de pertinência para mapear os valores das variáveis de entrada para graus de pertinência em cada conjunto. Por exemplo, a variável "carga atual" pode ser fuzzificada em conjuntos *Fuzzy* "baixa", "média" e "alta" usando funções de pertinência triangulares ou trapezoidais.

Para inferir a saída do modelo, operações como a de união, por exemplo, são aplicadas às regras para combinar os conjuntos *Fuzzy* resultantes de cada regra. O conjunto *Fuzzy* resultante é então defuzzificado para produzir um valor numérico que representa a carga futura antecipada.

A defuzzificação é a última etapa de um sistema *Fuzzy*, na qual um valor numérico é derivado do conjunto *Fuzzy* produzido pelas operações de inferência. O processo de defuzzificação envolve a conversão do conjunto *Fuzzy* em um valor numérico que representa uma solução definitiva. Existem inúmeros métodos de defuzzificação, cada um com suas próprias vantagens e desvantagens.

O método do centro de gravidade (ou centróide) é uma das técnicas de defuzificação mais prevalentes; ele calcula o ponto médio ponderado do conjunto *Fuzzy*. A seguinte equação pode ser usada para encontrar o centro de gravidade (CHIEN, 1990):

$$x_{cg} = \frac{\int_X u(x)xdx}{\int_X u(x)dx} \tag{2.9}$$

Onde:

$u(x)$: função de pertinência do conjunto fuzzzy;

X : é o universo do discurso;

X_{cg}: valor do centro de gravidade.

Em outras palavras, o centro de gravidade é a média ponderada dos valores do universo do discurso, com os pesos dados pelas funções de pertinência.

Outras técnicas de defuzzificação incluem o método da média, onde a média dos valores de pico do conjunto *Fuzzy* é calculada, e o método do máximo, onde o valor de pico do conjunto *Fuzzy* é selecionado como a saída. Cada técnica tem suas próprias vantagens e desvantagens, e a escolha depende da aplicação específica.

Alguns trabalhos utilizaram esta ferramenta para resolver o problema de previsão de carga. Entre eles, Pandian *et al.* (2006) apresentou uma metodologia de Lógica *Fuzzy* para resolver um problema típico de planejamento de sistema elétrico que é, não só prever a carga futura, mas também onde alocar os recursos para a expansão da rede. Três variáveis linguísticas para descrever 'distância' e cinco, para descrever 'preferência'. Chow e Tram (1996) afirmam que com o aumento de variáveis linguísticas implica no aumento da dimensão e da complexidade do problema. Neste mesmo trabalho, o "tempo" e a "temperatura" do dia são tomados como entradas para o controlador da lógica *Fuzzy* e a "carga prevista" é a saída. A variável de entrada "tempo" foi dividida em oito funções de associação triangular. Outra variável de entrada "temperatura" foi dividida em quatro funções de associação triangular. A "carga prevista" como saída foi dividida em oito funções de associação triangular. Estudos de caso foram realizados para a Unidade-II da Usina Termelétrica de Neyveli (NTPS-II) na Índia. Os valores de carga previstos difusos foram comparados com os valores previstos convencionais e a carga prevista correspondeu de perto à real dentro de $\pm$ 3%.

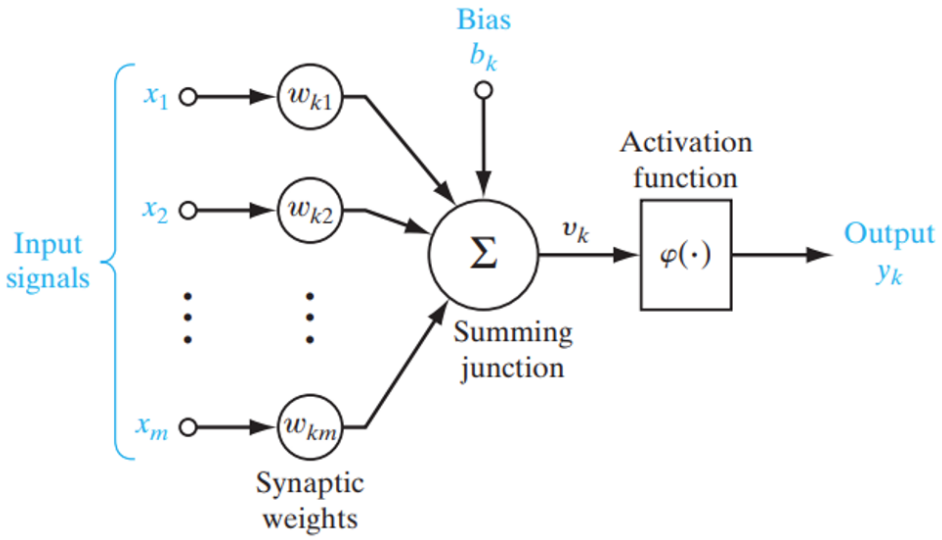
2.2.3.2 Redes Neurais Artificiais - RNA

Conforme visto anteriormente na Figura 6, as redes neurais são a técnica que mais profícua na literatura do segmento de previsores de carga. Isto porque funcionam bem mesmo com a natureza complexa e não linear dos dados de previsão de carga. As redes neurais podem lidar com várias fontes de incerteza, como variações sazonais, flutuações

de curto prazo e mudanças nas condições climáticas e econômicas. Além disso, aprendem automaticamente a partir de grandes quantidades de dados históricos, permitindo a melhoria contínua da precisão das previsões.

Possuem este nome por se basearem na estrutura fisiológica dos neurônios reais e de seu processamento paralelo e distribuído (BORDIGNON, 2012). Na Figura 9 é apresentada a estrutura de um neurônio artificial:

Figura 9 – Estrutura do neurônio artificial.



Fonte: Haykin (2009).

Da estrutura acima pode-se facilmente extrair o equacionamento:

$$v_k = \sum_{j=1}^{m} w_{kj} x_j + b_k$$

Em que o resultado do combinador linear v_k é o somatório dos produtos entre os sinais de entrada $\underline{\underline{X_i}}$ e seus respectivos pesos sinápticos W_{kj} somado a um viés b_k. Este viés é um escalar treinável que tem poder de aumentar ou diminuir o valor líquido do sinal de entrada (MANSUR, 2023). Portanto, se a soma ponderada das entradas for maior que o valor do bias, provocando um pulso na saída v_k e, consequentemente, a ativação do neurônio.

$$v_k = \sum_{j=1}^{m} w_{kj} x_j + b_k. \tag{2.10}$$

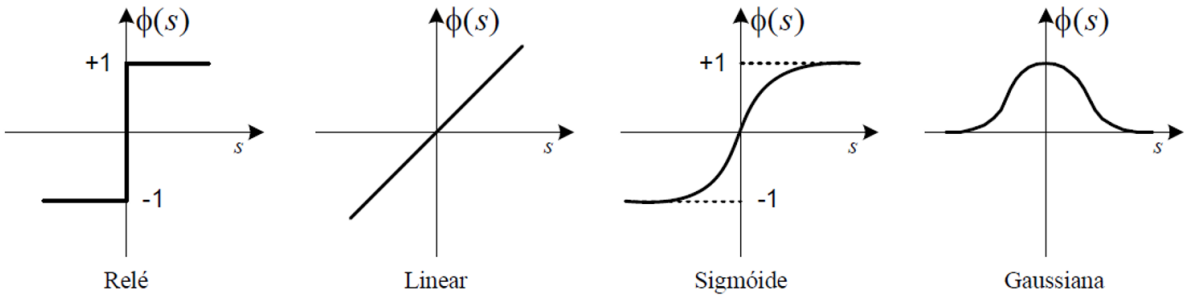
Em que o resultado do combinador linear v_k é o somatório dos produtos entre os sinais de entrada x_j e seus respectivos pesos sinápticos w_k somado a um viés b_k. Este viés é um escalar treinável que tem poder de aumentar ou diminuir o valor líquido do sinal de entrada (ALTRAN, 2010). Portanto, se a soma ponderada das entradas for maior que

o valor do bias, provocando um pulso na saída v_k e, consequentemente, a ativação do neurônio.

Uma vez ativado, a função de ativação é responsável por converter o sinal de entrada linear em um sinal de ativação não linear (MANSUR, 2023), permitindo a soluções de problemas desta natureza. Escolher a função de ativação de uma rede neural é uma etapa crucial para alcançar resultados de teste satisfatórios. Por determinar se um neurônio será ativado ou não, influencia diretamente na estrutura de dados de saída.

Ademais, conforme Haykin (2009), as funções de ativação normalizam o valor do sinal de ativação, limitando a amplitude da saída a intervalos finitos, como por exemplo, entre [0,1] ou [-1,1]. Na Figura 10 são apresentadas algumas das formas de funções de ativação mais comuns:

Figura 10 – Tipos de função de ativação.



Fonte: Altran (2010).

O processo de aprendizado se dá através do cálculo de erro das previsões do modelo frente o valor medido para, assim, quantificar os ajustes necessários nos pesos sinápticos e vieses da rede (por meio de um algoritmo de backpropagation) visando minimizar o erro e aperfeiçoar as previsões. Este erro é calculado por meio da função de custo, que é, principalmente nos casos de regressão, o cálculo do erro quadrático médio, que é dado por:

$$EQM = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\,, \tag{2.11}$$

onde n é o número de previsões, y_i: i-ésimo valor previsto e $\hat{y}_i$ é o i-ésimo valor medido.

Outro parâmetro relevante na concepção da rede neural para a resolução do problema é a arquitetura a ser utilizada. Basicamente, a estrutura de uma RNA é composta por uma camada de entrada, responsável pela recepção do sinal de entrada, uma camada intermediária ou oculta, responsável pelo processamento do sinal de entrada e que pode ser única ou várias em cascata e, finalmente, pela camada de saída, responsável por entregar resultados do processamento (BORDIGNON, 2012). Haykin, em (HAYKIN, 2009), identifica três classes de arquitetura de rede fundamentalmente diferentes:

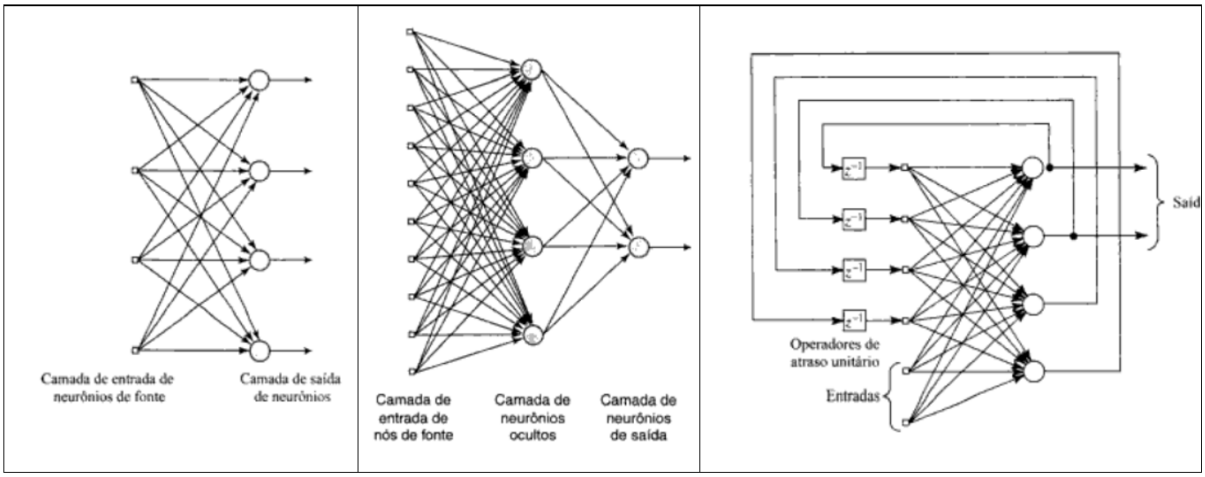
- Redes alimentadas adiante com camada única ou *feedfoward*: são redes em que a informação flui unidirecionalmente, sem realimentação, da camada de entrada para a camada de saída;

- Redes alimentadas diretamente com múltiplas camadas ou *perceptron* multicamadas (*Multi Layer perceptron* – MLP): são as redes *feedfoward* quando têm várias camadas de neurônios totalmente conectadas entre si;

- Redes neurais recorrentes – (*Recurrent Neural Network* – RNN): São redes que possuem laços de realimentação, permitindo que informações sejam armazenadas e utilizadas para influenciar as saídas em momentos futuros. Por isso, permitem a modelagem de séries temporais. Duas topologias de RNN são a *Long Short-Term Memory* (LSTM) e a *Gated Recurrent Unit* (GRU).

A seguir, uma ilustração esquemática das arquiteturas das redes neurais supramencionadas:

Figura 11 – Esquemas das redes *feedfoward*, MLP e RNN, respectivamente.

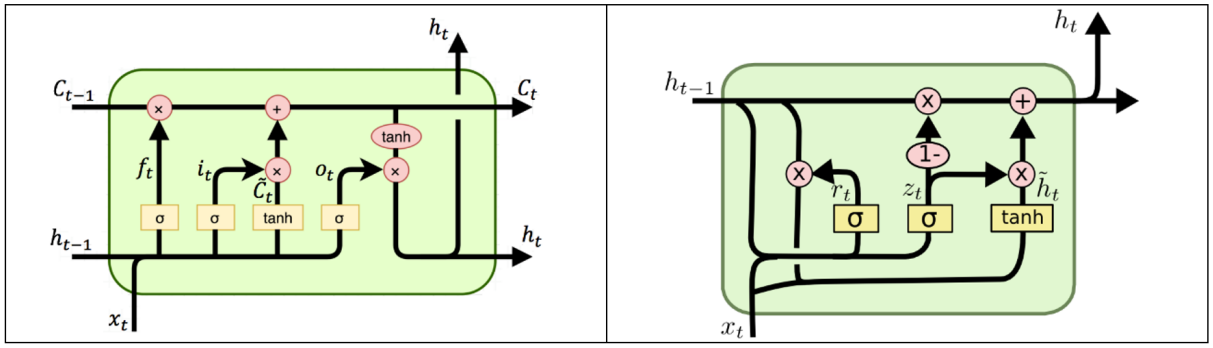


Fonte: Haykin (2001).

Com relação às Redes Neurais Recorrentes, as RNNs têm dificuldades em manter informações de longo prazo, o que pode levar a problemas de explosão do gradiente. Para lidar com esses problemas, foram propostas as variantes de RNNs, a LSTM (*Long Short-Term Memory*) e a GRU (*Gated Recurrent Unit*). A Figura 12 apresenta a estrutura destas duas topologias derivadas das RNN:

A LSTM é uma arquitetura de rede neural recorrente que foi projetada para permitir que a rede aprenda e mantenha informações de longo prazo, em virtude da sua estrutura contendo três portões ou chaves (*gates*), a célula tradicional, a célula de entrada e de saída e a função de esquecimento, definindo as informações que devem ser retidas (CUNHA, 2022).

Figura 12 – Estruturas das RNNs LSTM e GRU, respectivamente.



Fonte: Olah (2015).

A GRU é outra arquitetura de rede neural recorrente que foi desenvolvida para resolver os mesmos problemas das RNNs originais. A estrutura da GRU se difere da LSTM nos *gates*, possuindo funções de atualização e *reset*. Enquanto a LSTM permite que seja selecionada a quantidade de informações que devem ser passadas adiante, a GRU filtra os dados a serem mantidos ou negligenciados (CUNHA, 2022).

Entre alguns trabalhos que utilizaram de redes neurais para realizar predição de carga, Park *et al.* (1991) utilizou uma RNA para aprender a relação entre temperaturas e cargas passadas, atuais e futuras. Para fornecer a carga prevista, a ANN interpolou entre os dados de carga e temperatura em um conjunto de dados de treinamento, apresentando um erro médio de 4,22% para previsões de 24 horas a frente. Em Lu, Wu e Vemuri (1993) foi utilizado uma rede RNA *feedfoward* com dados horários de carga e temperatura por 12 meses, sendo capazes de prever com erro médio de 1.24% a carga de um dia. Em Kong *et al.* (2017) é proposto uma rede LSTM para prever uma carga residencial inserido num sistema de *Smart Grid* para um horizonte de curto prazo. Como resultado, a abordagem LSTM adotada superou algoritmos de referência.

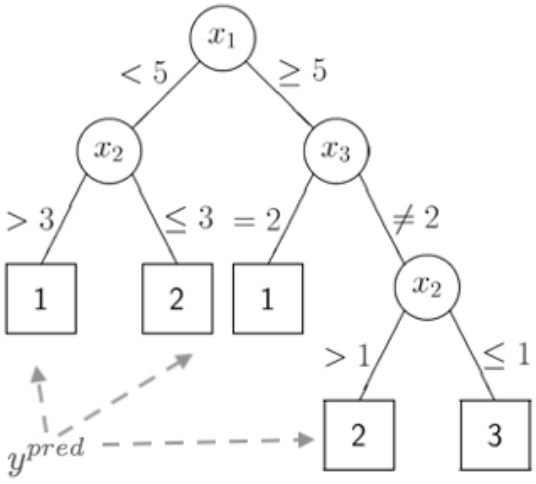
2.2.3.3 Modelos baseados em Árvore de Decisão

As Árvores de Decisão são um dos métodos mais comuns e eficazes em aprendizado de máquina, sendo aplicáveis tanto em problemas de classificação quanto de regressão. O modelo é construído através de uma série de perguntas, feitas de maneira sequencial, que resultam em uma "decisão". Esta decisão pode ser uma classe em problemas de classificação, ou um valor contínuo em problemas de regressão. A estrutura do modelo se assemelha a uma árvore, na qual cada nó interno representa uma pergunta ou teste de algum atributo, cada ramo é um resultado desse teste, e cada nó folha é uma decisão ou previsão.

A Árvore de Decisão em si é uma representação gráfica de possíveis soluções para uma decisão com base em uma sequência de condições. Ela começa com um único nó, chamado de raiz, que se divide em possíveis resultados. Cada um desses resultados leva

a nós adicionais, que se ramificam em outras possibilidades. Este processo continua até que se chegue a um nó folha, que é um nó terminal sem ramificações, representando uma previsão ou uma decisão, conforme ilustra a Figura 13.

Figura 13 – Esquema de uma Árvore de Decisão aplicada à regressão.



Fonte: Teixeira-Pinto e Harezlak (2022).

A árvore é construída de forma indutiva, de cima para baixo, a partir dos dados. O algoritmo divide os dados em subconjuntos de maneira recursiva, de modo a obter subconjuntos cada vez mais puros. A "pureza"é avaliada usando uma métrica, que pode ser a entropia, o índice Gini ou a redução de variância, dependendo se o problema é de classificação ou regressão.

Para problemas de regressão, o método mais comumente utilizado em árvores de decisão é a de redução da variância. Matematicamente, a variância (Var) para um conjunto de dados D com N observações pode ser calculada como:

$$Var(D) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \bar{y})^2 \tag{2.12}$$

Onde y_i é o valor da variável alvo para a i-ésima observação e $\bar{y}$ é a média da variável alvo sobre todas as observações.

Para uma divisão de D em dois subconjuntos D_1 e D_2, a redução da variância causada por essa divisão é definida como a diminuição total da variância nos subconjuntos, ponderada pelo tamanho dos subconjuntos, ou seja:

$$Reduction\,(D, D_1, D_2) = Var(D) - \left(\frac{|D_1|}{|D|}Var\,(D_1) + \frac{|D_2|}{|D|}Var\,(D_2)\right) \tag{2.13}$$

Onde $|D|, |D_1|$ e $|D_2|$ representam o número de observações em D, D_1 e D_2, respectivamente.

Para cada potencial ponto de divisão em cada variável de entrada, a árvore de decisão avalia a redução da variância, optando pela divisão que maximiza essa redução. Este procedimento é iterado recursivamente até que se alcance um critério de parada, que pode ser um número mínimo de observações em cada nó folha, uma redução mínima da variância, um número máximo de níveis na árvore, ou até que nenhuma divisão adicional resulte em uma melhoria significativa na medida de impureza.

As árvores de decisão oferecem um método intuitivo e flexível para aprender padrões nos dados, lidando bem com variáveis categóricas e contínuas, e fornecendo visualizações claras das regras de decisão que elas aprendem. No entanto, elas têm uma propensão para se ajustar demais aos dados de treinamento e não generalizam bem para dados não vistos, um problema conhecido como *overfitting*. A maneira mais comum de mitigar o *overfitting* nas árvores de decisão é através de técnicas de conjunto, que combinam a previsão de várias árvores de decisão para fazer uma previsão final. Uma dessas técnicas é conhecida como Floresta Aleatória (*Random Forest*).

O ***Random Forest*** é um algoritmo de aprendizado de máquina que cria um conjunto, ou "floresta", de árvores de decisão independentes durante o treinamento e gera previsões usando a moda (para classificação) ou a média (para regressão) das previsões de cada árvore individual. A ideia principal por trás das Florestas Aleatórias é que um conjunto de árvores de decisão relativamente descorrelacionadas operando em conjunto terá um desempenho superior a qualquer uma das árvores individuais.

Nesse sentido, a Floresta Aleatória implementa o conceito de *bagging* (*bootstrap aggregating*), uma técnica de aprendizado de conjunto que visa a redução da variância. O *bagging* é um procedimento geral que pode ser usado para reduzir a variância para um conjunto de árvores de decisão. Ele opera gerando múltiplas versões de um preditor e usando esses para obter uma saída agregada. No caso da *Random Forest*, as árvores de decisão são os preditores. Durante o treinamento, as árvores são criadas usando subconjuntos dos dados de treinamento selecionados aleatoriamente com reposição, o que significa que algumas observações podem ser repetidas na amostra.

Cada árvore na floresta aleatória é construída de forma a maximizar a pureza dos nós, de acordo com a função de impureza. No entanto, para evitar a correlação entre as árvores, em cada divisão de um nó, a *Random Forest* considera apenas um subconjunto de atributos selecionado aleatoriamente, em vez de procurar as melhores divisões entre todos os atributos.

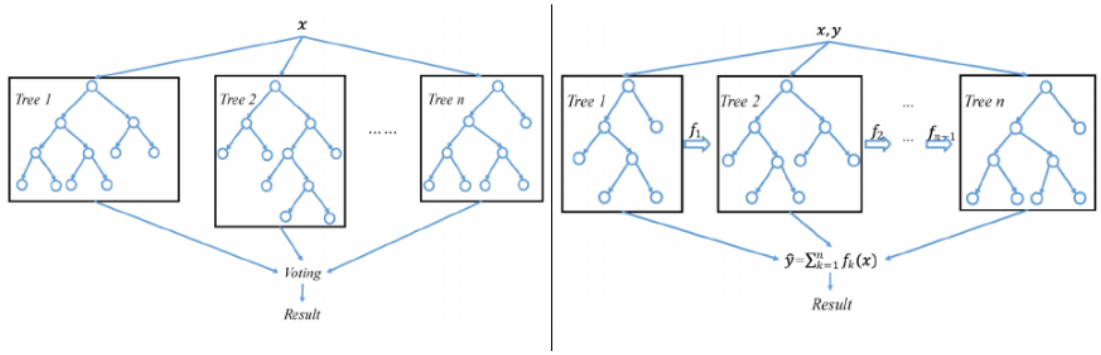
No contexto da regressão, assim como na Árvore de Decisão, a *Random Forest* utiliza a redução de variância como critério de divisão. Nesse cenário, a árvore tenta dividir

as características de forma que os grupos resultantes tenham a menor variância de valores possível, isto é, que os valores de cada grupo sejam o mais semelhantes possível entre si. O objetivo é minimizar a função de custo, que é a soma das variâncias dos grupos formados pela divisão.

Além do *bagging*, existem outras técnicas de aprendizado de conjunto como o *boosting*. O *Boosting* também combina vários modelos fracos para criar um modelo forte, no entanto, em vez de construir todos os modelos independentemente e de forma paralela (como no *bagging*), no *boosting* os modelos são construídos de forma sequencial, onde cada novo modelo tenta corrigir os erros cometidos pelo modelo anterior.

Aqui é onde se encontra a ligação entre a *Random Forest* e o ***XGBoost***. O *XGBoost* (*eXtreme Gradient Boosting*) é uma implementação avançada e eficiente do algoritmo de *boosting*, que constrói o modelo de forma aditiva e sequencial, minimizando ao mesmo tempo uma função de perda com a adição de uma regularização para evitar o *overfitting*. Diferentemente da *Random Forest*, que gera previsões usando várias árvores de decisão que foram treinadas independentemente, o *XGBoost* gera previsões usando uma sequência de árvores de decisão, onde cada nova árvore ajuda a corrigir erros cometidos pelas árvores anteriores. A Figura 14 ilustra esquematicamente a diferença estrutural entre as duas técnicas.

Figura 14 – Esquema de Regressão por *Bagging* e *Boosting*.



Fonte: Wang *et al.* (2019).

No caso de problemas de regressão, o *XGBoost* usa o gradiente descendente para minimizar a diferença entre as previsões e os valores reais, ajustando o modelo aditivamente para melhorar gradualmente a sua acurácia. Isso é feito construindo cada árvore de tal forma que ela compensa os resíduos dos erros das árvores anteriores na sequência, de forma que a soma dos resíduos seja minimizada.

O *XGBoost* também usa a estrutura de árvore para a aprendizagem de modelos, mas faz isso de uma maneira diferente da *Random Forest*. Enquanto a *Random Forest* constrói uma floresta de árvores de decisão de maneira independente e paralela, o *XGBoost* faz isso de forma sequencial e aditiva. Este aspecto sequencial do *XGBoost* faz com que ele

possa ser mais efetivo em reduzir o erro de previsão, à custa de um tempo de treinamento potencialmente mais longo e uma maior complexidade computacional.

3 ESTADO-DA-ARTE

Para fins de organização, serão apresentados o estado da arte dos preditores de carga, serparados por preditores de curto, médio e longo prazo:

3.1 Previsores de carga de curto prazo

Na literatura encontramos diversos preditores de curto prazo, ou seja, capazes de prever o comportamento da carga ao longo de 24 horas ou até mesmo poucas horas a frente, os chamados preditores STLF (de *Short Term Load Forecaster*). A seguir, alguns dos mais atuais e relevantes STLF são mencionados:

Em Guo *et al.* (2021) é apresentado três métodos para realizar a previsão de carga: SVM, o método de regressão por Floresta Aleatória e uma LSTM. O trabalho integrou algumas das vantagens das três filosofias aliadas a uma técnica de pré-processamento a fim de melhorar o resultado preditivo. No final, realizaram um teste comparativo com base em dados reais e concluíram que a proposta combinada foi capaz de alcançar uma precisão relativamente alta.

Rafi *et al.* (2021) propôs uma técnica baseada na integração de uma rede neural convolucional (CNN) e uma LSTM, aplicado ao sistema de energia de Bangladesh para fornecer previsões de curto prazo da carga elétrica. Além disso, a eficácia da técnica proposta foi validada pela comparação dos erros de previsão com a de algumas abordagens existentes, como a LSTM individualmente, a rede de função de base radial e algoritmo de aumento de gradiente extremo. Verificou-se que a estratégia proposta resulta em maior precisão e acurácia na previsão de carga de curto prazo.

Em Lin *et al.* (2022) foi proposto uma rede LSTM baseada em uma metodologia de atenção de duas etapas. Na primeira, é construído um codificador baseado em atenção de recursos para calcular a correlação das características de entrada com a carga elétrica em cada intervalo de tempo. As características de entrada mais relevantes foram selecionadas de forma adaptativa. Na segunda etapa, um decodificador baseado em atenção temporal foi desenvolvido para explorar as dependências temporais. Em seguida, um modelo LSTM integra esses resultados de atenção e as previsões probabilísticas foram obtidas usando uma função de perda. A eficácia do método proposto foi verificada através do conjunto aberto, mostrando maior precisão e maior habilidade de generalização em relação a outros modelos de previsão de última geração.

Percebe-se, assim, que a utilização da LSTM, individualmente como combinada com outras metodologias, tem sido objeto dos estudos mais recentes e relevantes no segmento de STLF, obtendo bons resultados práticos.

3.2 Previsores de carga de médio prazo

O conceito de previsor de médio prazo varia um pouco de autor para autor mas, via de regra, o MTLF (de *Mid-Term Load Forecaster*) é aquele capaz de prever o perfil da carga num intervalo de semanas a meses à frente. Alguns MTLF mais recentes e citados na literatura são apresentados a seguir:

Baek (2019) apresenta um método de previsão da ponta de carga diária de médio prazo utilizando rede neural artificial recorrente (RANN). O artigo discutiu sobre a dificuldade de aprendizado e estimativa de previsão de carga de longo e médio prazo devido à falta de dados de treinamento e ao aumento de erros acumulados na estimativa de longo período, propondo uma estrutura de previsão de carga de médio prazo capaz de superar esses problemas por meio da substituição de dados de entrada para dias especiais e uma aplicação da RANN. Além disso, a RANN proposta apresentou boas performances na estimativa de aumento repentino e não linear da demanda durante ondas de calor. Os resultados de estudos de caso usando dados de carga da Coreia do Sul foram apresentados para mostrar as performances e a eficácia da proposta.

Askari e Keynia (2020) introduz em seu texto que o problema de previsão de médio prazo, por ter um comportamento não estacionário, volátil e não linear, apresentam sérios desafios. Por outro lado, justifica a não utilização de muitas outras variáveis e parâmetros relativos por considerar que estes podem afetar o padrão da carga, para previsores de médio prazo. Portanto, apresenta um novo método composto com base em uma rede neural de *perceptron* de várias camadas e técnicas de otimização para resolver o problema MTLF. O método proposto tem um algoritmo de treinamento ótimo composto por dois algoritmos de busca (Otimização do 'Enxame de Partículas' (*Particle Swarm Optization* - PSO) e Otimizador do Formiga-Leão (*Improved Ant-Lion optimization* - ALO) aprimorado) e uma rede neural de *perceptron* de várias camadas. Através dos testes de Erro Médio Quadrático (*Mean Square Error* – MSE) e de Erro Médio Absoluto Percentual (*Mean Absolute Percentage Error* – MAPE), atestaram que o método tem alta precisão. Por isso, apresentam o trabalho como um novo método de otimização, pela sua capacidade de treinamento dos pesos do preditor.

Dudek e Pełka (2021) propuseram o uso de padrões de sequências de séries temporais para representação de séries temporais, de forma a garantir a unificação dos dados de entrada e saída por meio da filtragem de tendência e equalização de variância, simplificando problema de previsão, permitindo a utilização de modelos baseados em similaridade de padrões. Para isso, foram utilizados modelos *k-nearest neighbors* (KNN), *Fuzzy Neighborhood*, Regressão Kernel e uma rede neural regressiva. Propuseram três variantes da abordagem. Uma básica e duas soluções híbridas combinando métodos baseados em similaridade e métodos estatísticos (ARIMA e suavização exponencial). Na parte experimental do trabalho, os modelos propostos foram utilizados para prever a demanda mensal de eletricidade

em 35 países europeus. Os resultados mostram o alto desempenho dos modelos propostos, que superam tanto os modelos estatísticos clássicos comparativos quanto os modelos de aprendizado de máquina em termos de precisão, simplicidade e facilidade de otimização.

3.3 Previsores de carga de longo prazo

Previsores de carga de longo prazo são aqueles capazes de prever a demanda de energia além de um ano a frente e tem um caráter de planejamento ótimo do sistema.

Nalcaci, Özmen e Weber (2019) propuseram três modelos baseados em *splines* de regressão adaptativa multivariável (MARS), rede neural artificial (ANN) e regressão linear (LR) para modelar a carga elétrica em geral na rede de distribuição de eletricidade turca, não só por meio de previsões de longo prazo, mas também de médio e curto prazo. Esses modelos preveem a relação entre a demanda de carga e várias variáveis ambientais: vento, umidade, carga do dia do ano (feriado, verão, dia útil, etc.) e dados de temperatura. Ao comparar esses modelos, mostramos que o modelo MARS oferece resultados mais precisos e estáveis do que os modelos ANN e LR.

Wang, Du e Wang (2020) apresenta uma abordagem baseada em LSTM para prever o consumo de energia periódica. Primeiramente, são filtrados alguns padrões ocultos pelo gráfico de autocorrelação entre os dados medidos. A análise de correlação e a análise de mecanismo contribuem para encontrar as variáveis secundárias apropriadas como entrada do modelo. Além disso, a variável de tempo é complementada para capturar precisamente a periodicidade. Em seguida, uma rede LSTM é construída para modelar e prever dados sequenciais. Os resultados experimentais demonstram que o método proposto tem um desempenho de previsão mais elevado em comparação com diversos métodos de previsão tradicionais, como o modelo de média móvel autoregressivo (ARMA), o modelo de média móvel autoregressivo fracionário integrado (ARFIMA) e a rede neural de retropropagação (BPNN). O RMSE do LSTM é 19,7%, 54,85% e 64,59% menor do que o BPNN, ARMA e ARFIMA nos dados de teste.

3.4 Considerações em relação ao estado da arte

A pesquisa do estado-da-arte de previsores de carga apontou que existe grande interesse no estudo de STLF, dada a grande disponibilidade de literatura encontrada para este segmento. Sem realizar uma avaliação estatística profunda dos trabalhos disponíveis, é possível apontar que boa parte dos trabalhos mais relevantes tem convergido para a utilização da técnica de LSTM para resolver o problema de previsão de curto prazo.

Para os MTLF, não há tantos trabalhos disponíveis quanto o STLF. Outra observação é que, neste segmento, não há uma convergência para um método consagrado para a resolução do problema. Entretanto, majoritariamente entre os trabalhos citados,

para a previsão de carga de médio prazo, a solução mais comum tem sido a utilização de combinação de diversas técnicas. Os resultados têm se demonstrado satisfatórios.

Já para o problema de previsão de carga à longo prazo, não há tanta literatura recente quanto para os segmentos anteriores. Realmente é bem desafiador realizar previsão de carga anos à frente, uma vez que esta variável é fortemente dependente de fatores econômicos – e até sanitários, como foi o caso da redução da capacidade econômica de vários países decorrentes da pandemia de Covid-19. Neste contexto de LTLF, modelos regressivo que se utilizam de séries temporais tem obtidos maior destaque na literatura recente.

4 METODOLOGIA

Neste capítulo será descrito a metodologia utilizada para desenvolver o modelo de previsão de carga. Será descrito o processo de aquisição, tratamento dos dados, a inclusão de novas *features*, agregando características além da própria carga medida à base de dados, até a consolidação do *dataframe* utilizado como estudo de caso.

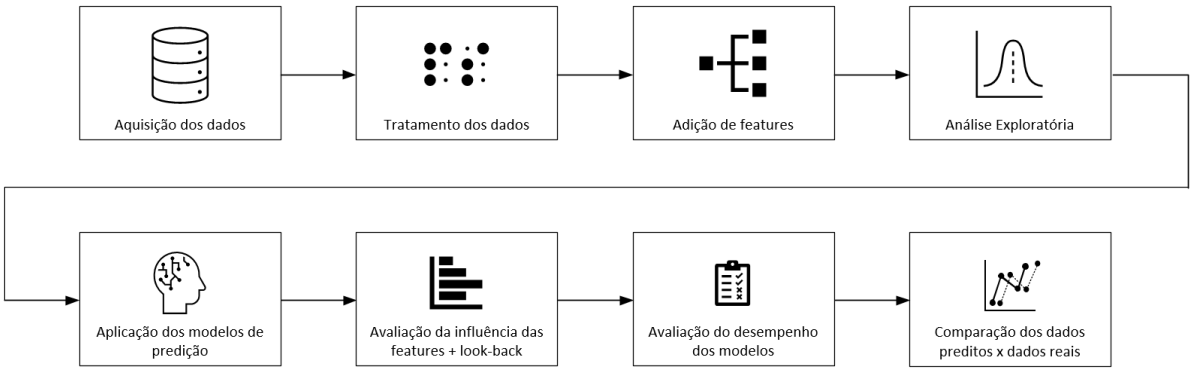
Serão apresentados os testes de análise exploratória na base de dados, buscando entender as tendências, padrões e correlações presentes nos dados e, assim, caracterizar a série temporal, de forma a facilitar os ajustes dos modelos preditivos.

Em seguida, serão apresentados os modelos de previsão aos quais os dados foram aplicados. Nesse estágio, foi importante para o desenvolvimento dos modelos preditores realizar algumas avaliações específicas, tais como, a importância da escolha adequada do método de separação das amostras em conjunto de treinamento e teste, a influência das *features* e do método de *look-back* no processo de previsão dos modelos.

Finalmente, serão apresentadas as métricas utilizadas para a avaliação do desempenho dos modelos, a comparação entre os dados preditos e os dados reais de cada modelo e os critério utilizado para qualificar os melhores modelos para produção.

Na Figura 15 é apresentado o *workflow* adotado neste trabalho:

Figura 15 – Etapas do desenvolvimento do preditor de carga proposto.



Fonte: do próprio autor.

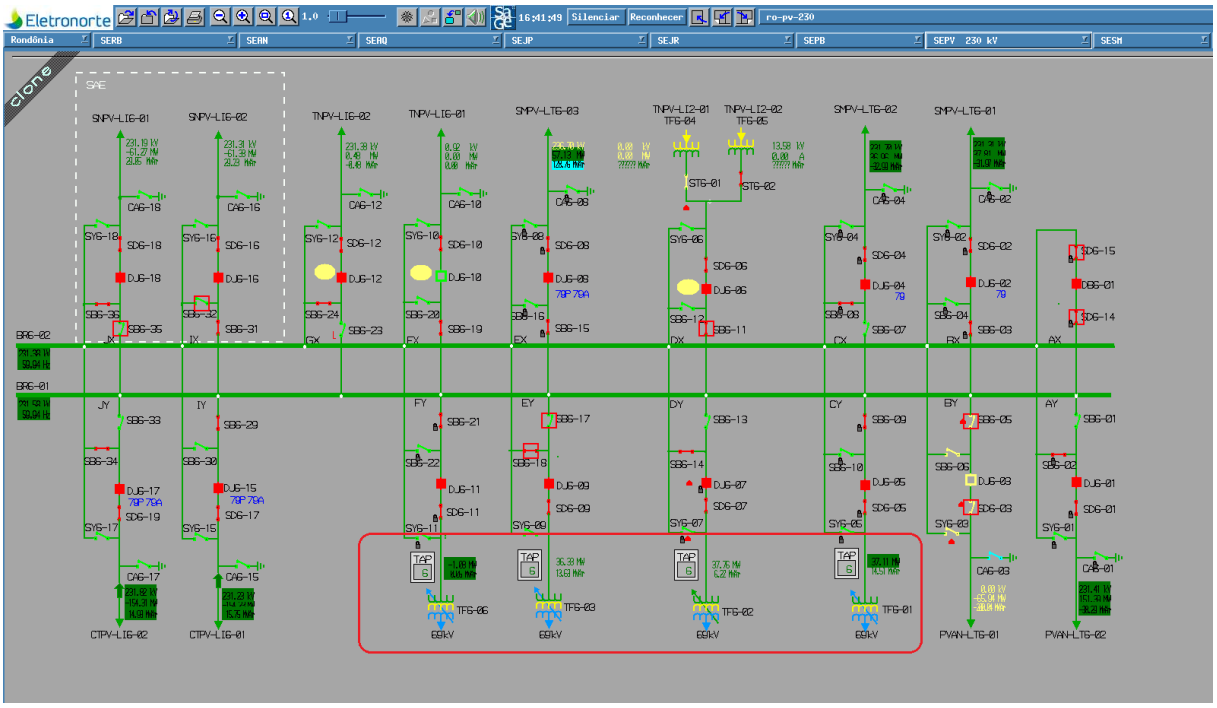
4.1 Aquisição dos dados

As medidas de carga (medições de potência ativa, em MW) escolhidas para formar o conjunto de dados para realizar o treinamento e o teste dos modelos preditores deste trabalho foram as da subestação SE Porto Velho, sob concessão da Eletrobras Eletronorte. Esta subestação foi escolhida por ser considerada uma instalação estratégica, uma vez

que seus quatro transformadores 230/69/13.8 kV de 100 MVA cada um são responsáveis por atender toda uma capital. Outro critério adotado foi a baixíssima frequência de desligamentos intempestivos, minimizando a ocorrência de amostras nulas durante o período de medição escolhido.

Os dados utilizados no trabalho foram obtidos por meio do acesso ao banco de dados de medição do SAGE[1], referente a leitura de potência ativa de cada transformador da SE Porto Velho, entre junho de 2022 até abril de 2023. Cada amostra do *dataframe* obtido (dados brutos) corresponde ao valor médio das amostras instantâneas, integralizados numa janela de 5 minutos. Ou seja, O conjunto de dados utilizados neste trabalho corresponde a uma janela temporal de 11 meses com uma taxa de amostragem de 5 minutos, o que corresponde a um conjunto bruto de dados de aproximadamente 96 mil amostras de medidas de potência ativa, lidas no enrolamento primário de cada transformador. Na Figura 16 é apresentada a tela de exibição da SE Porto Velho no sistema supervisório do SAGE, enquanto a Tabela 2 ilustra o formato bruto dos dados adquiridos do banco de dados do SAGE.

Figura 16 – SE Porto Velho no sistema SAGE.



Fonte: Eletrobras Eletronorte. Destaque em vermelho do autor.

[1] O SAGE é um sistema SCADA/EMS (*Supervisory Control and Data Acquisition/Energy Management System*) de grande porte e alto desempenho, desenvolvido e atualizado pelo Centro de Pesquisas de Energia Elétrica da Eletrobras, o CEPEL. É utilizado por concessionárias de geração, transmissão e distribuição de energia elétrica no Brasil, em especial as empresas do grupo Eletrobras, além do Operador Nacional do Sistema Elétrico (ONS), em todos os seus centros de controle. (CEPEL, 2023)

Tabela 2 – Fragmento da tabela de medidas adquiridas do banco de dados do SAGE para o transformador PVTF6-06.

"id"	"dthr"	"min"	"min_dthr"	"max"	"max_dthr"	"valor"	"valor_dthr"	"media"
"PVTF601MP6"	"2022-06-03 21:05:00-03"	46.383327	"2022-06-03 21:09:25-03"	46.514217	"2022-06-03 21:09:01-03"	46.406853	"2022-06-03 21:02:45-03"	46.429953956604
"PVTF601MP6"	"2022-06-03 21:00:00-03"	46.22537	"2022-06-03 21:01:31-03"	46.528095	"2022-06-03 21:02:15-03"	46.4166	"2022-06-03 21:00:00-03"	46.41399264017741
"PVTF601MP6"	"2022-06-03 21:10:00-03"	46.383327	"2022-06-03 21:09:25-03"	46.53897	"2022-06-03 21:12:29-03"	46.383327	"2022-06-03 21:09:25-03"	46.44106179555257
"PVTF601MP6"	"2022-06-03 21:15:00-03"	46.401596	"2022-06-03 21:14:15-03"	46.537697	"2022-06-03 21:18:45-03"	46.401596	"2022-06-03 21:14:15-03"	46.43562126159668
"PVTF601MP6"	"2022-06-03 21:20:00-03"	46.537697	"2022-06-03 21:18:45-03"	46.537697	"2022-06-03 21:18:45-03"	46.537697	"2022-06-03 21:18:45-03"	46.537696838378906
"PVTF601MP6"	"2022-06-03 21:25:00-03"	46.537697	"2022-06-03 21:18:45-03"	46.660454	"2022-06-03 21:27:12-03"	46.537697	"2022-06-03 21:18:45-03"	46.605420951843264
"PVTF601MP6"	"2022-06-03 21:30:00-03"	46.558475	"2022-06-03 21:29:57-03"	46.558475	"2022-06-03 21:29:57-03"	46.558475	"2022-06-03 21:29:57-03"	46.558475494384766
"PVTF601MP6"	"2022-06-03 21:35:00-03"	46.558475	"2022-06-03 21:29:57-03"	46.558475	"2022-06-03 21:29:57-03"	46.558475	"2022-06-03 21:29:57-03"	46.558475494384766
"PVTF601MP6"	"2022-06-03 21:40:00-03"	46.410347	"2022-06-03 21:40:51-03"	46.69007	"2022-06-03 21:42:55-03"	46.558475	"2022-06-03 21:29:57-03"	46.559561042785646
"PVTF601MP6"	"2022-06-03 21:45:00-03"	46.561325	"2022-06-03 21:47:31-03"	46.82419	"2022-06-03 21:49:53-03"	46.56982	"2022-06-03 21:44:44-03"	46.64932912190755
"PVTF601MP6"	"2022-06-03 21:50:00-03"	46.644405	"2022-06-03 21:53:01-03"	46.82419	"2022-06-03 21:49:53-03"	46.82419	"2022-06-03 21:49:53-03"	46.75287436167399
"PVTF601MP6"	"2022-06-03 21:55:00-03"	46.644405	"2022-06-03 21:53:01-03"	46.644405	"2022-06-03 21:53:01-03"	46.644405	"2022-06-03 21:53:01-03"	46.644405364990234
"PVTF601MP6"	"2022-06-03 22:00:00-03"	46.58674	"2022-06-03 22:00:00-03"	46.756607	"2022-06-03 22:03:20-03"	46.58674	"2022-06-03 22:00:00-03"	46.64336140950521
"PVTF601MP6"	"2022-06-03 22:05:00-03"	46.756607	"2022-06-03 22:03:20-03"	46.756607	"2022-06-03 22:03:20-03"	46.756607	"2022-06-03 22:03:20-03"	46.75660705566406
"PVTF601MP6"	"2022-06-03 22:10:00-03"	46.6314	"2022-06-03 22:13:22-03"	46.795773	"2022-06-03 22:12:50-03"	46.756607	"2022-06-03 22:03:20-03"	46.67197224934896
"PVTF601MP6"	"2022-06-03 22:15:00-03"	46.6314	"2022-06-03 22:13:22-03"	46.6314	"2022-06-03 22:13:22-03"	46.6314	"2022-06-03 22:13:22-03"	46.63140106201172
"PVTF601MP6"	"2022-06-03 22:20:00-03"	46.6314	"2022-06-03 22:13:22-03"	46.817772	"2022-06-03 22:21:41-03"	46.6314	"2022-06-03 22:13:22-03"	46.75502705891927
"PVTF601MP6"	"2022-06-03 22:25:00-03"	46.692097	"2022-06-03 22:28:36-03"	46.817772	"2022-06-03 22:21:41-03"	46.817772	"2022-06-03 22:21:41-03"	46.79916703542074
"PVTF601MP6"	"2022-06-03 22:30:00-03"	46.793633	"2022-06-03 22:29:11-03"	47.03158	"2022-06-03 22:34:35-03"	46.793633	"2022-06-03 22:29:11-03"	46.91285223642985
"PVTF601MP6"	"2022-06-03 22:35:00-03"	47.03158	"2022-06-03 22:34:35-03"	48.22752	"2022-06-03 22:39:26-03"	47.03158	"2022-06-03 22:34:35-03"	47.55956553141276

Fonte: Eletrobras Eletronorte

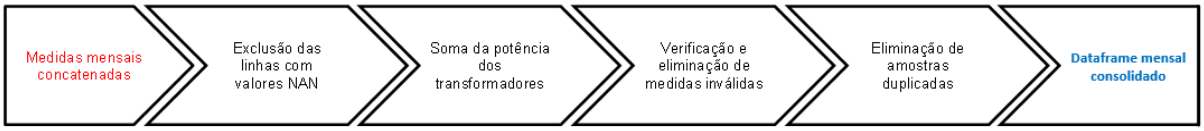
4.2 Tratamento dos Dados

Após a aquisição das medidas de potência ativa de cada um dos 4 transformadores da SE Porto Velho, mês a mês, deu-se inicio ao processo de tratamento dos dados.

Inicialmente, foram extraídas as colunas mais importantes do conjunto de dados bruto, apresentada na Tabela 2: as colunas 'dthr', que corresponde à coluna com informação da data e hora da amostra integralizada em 5 minutos e 'media', correspondente à media das amostras dentro desta mesma janela. Este processo foi realizado para os quatro transformadores dentro do mesmo mês. Então, estes dados foram concatenados entre si num mesmo *dataframe*, de forma a obter uma tabela mensal com a medida de carregamento em cada transformador.

Estando pronto o *dataframe* mensal com o conjunto de medidas de carga cada transformador, deu-se inicio ao processo de limpeza e tratamento dos dados. O processo adotado seguiu o mecanismo ilustrado pela Figura 17:

Figura 17 – Processo de tratamento dos dados adquiridos.



Fonte: do próprio autor.

O processo de exclusão das linhas com valores 'NAN' (*not a number*) visou eliminar simultaneamente as amostras de leitura de carga dos transformadores quando pelo menos um deles, por alguma falha de medição, apresentou um valor inexistente. Este tratamento possibilitou o processo seguinte, da soma medida das cargas dos transformadores, uma vez que a soma de uma amostra válida com uma com valor 'NAN' é inviável ou discrepante.

Em seguida, foi criada uma nova coluna no *dataframe*, resultado da soma da carga dos quatro transformadores, indicando a carga total da instalação naquele dado instante. A esta nova coluna foi atribuída a característica 'CARGA-PV'. Este será o objeto sobre o qual serão aplicados os modelos de previsão, posteriormente.

Após a criação do atributo que corresponde ao carregamento total da instalação, verificou-se a existência e eliminação de medidas inválidas, por uma eventual falha de medição. Medidas com valor menor que zero, ou com valores muito acima à capacidade máxima de sobrecarga dos transformadores, indicando uma medição irreal.

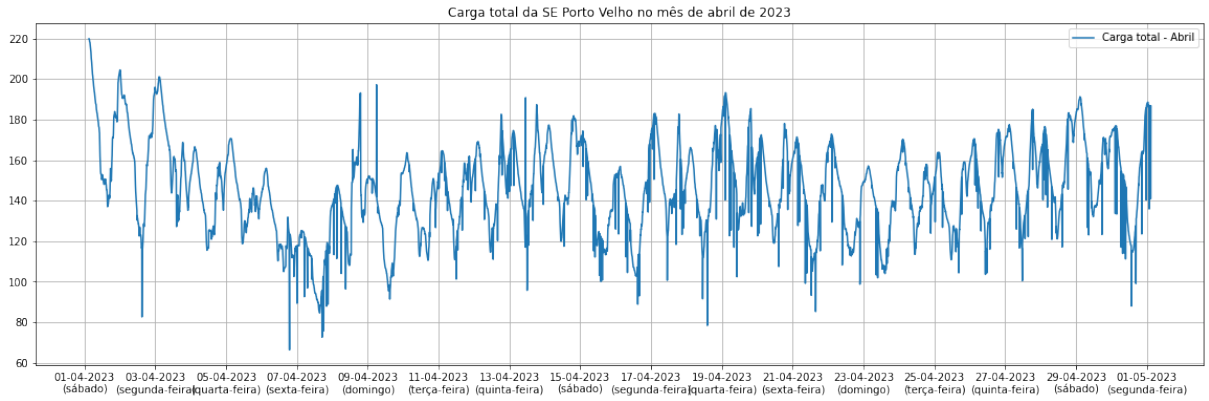
Finalmente, as amostras que apresentaram duplicidades sequenciais ou repetições mantidas ao longo do tempo também foram eliminadas, por caracterizarem algum tipo de travamento do sistema de medição, tornando-se, assim inválidas.

Toda essa eliminação de medidas inválidas não afetam a qualidade do conjunto de

dados, uma vez que a taxa de amostragem das medidas é muito baixa comparada com a janela temporal do *dataframe* ainda mensal.

Portanto, após todo o tratamento descrito acima, foi obtido um *dataframe* consolidado, capaz de descrever a variação da carga da subestação ao longo de um mês. A Figura 18 ilustra o conteúdo do *dataframe*:

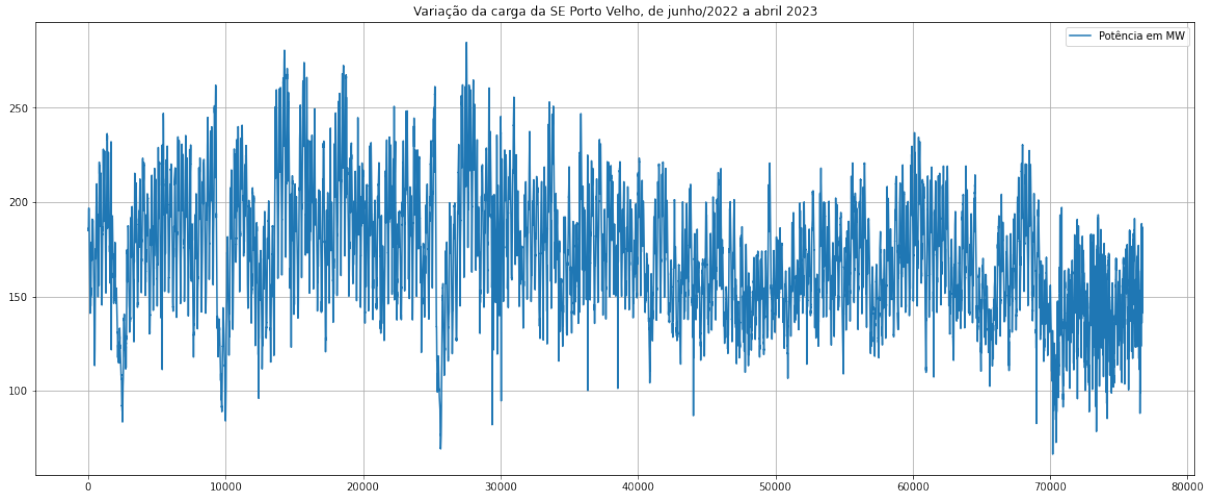
Figura 18 – Variação da carga ao longo do mês de abril/2023 na SE Porto Velho.



Fonte: do próprio autor.

Finalmente, após repetir este processo para cada mês, todos estes subconjunto de dados foram concatenados entre si, de forma a obter um único *dataframe* com os dados tratados e consolidados, abrangendo o intervalo entre junho de 2022 a abril de 2023. Este *dataframe* é a primeira versão do conjunto completo de dados de carga que, posteriormente, será utilizado no processo de predição, contendo 76684 objetos. A Figura 19 ilustra o conteúdo final da carga utilizada nos treinamentos e testes dos modelos:

Figura 19 – Série temporal consolidada para treinamento e teste dos modelos de predição. Carga da SE Porto Velho entre junho/22 e abril/2023



Fonte: do próprio autor.

4.3 Adição de *features* ao conjunto de dados

Conforme mencionado anteriormente no capítulo 2, fatores como feriados e finais de semana influenciam o perfil de consumo da energia e, portanto, possuem uma característica majoritariamente mais leve em comparação a um dia útil. Adicionalmente, desde a primeira metade do século XX, há literatura indicando a relação entre fatores climáticos e o perfil de carga de uma região Hong e Wang (2022). Alguns exemplos são Park *et al.* (1991) e Lu, Wu e Vemuri (1993), citados no capítulo 2 e Nalcaci, Özmen e Weber (2019), citado no capítulo 3. Desse modo, uma boa prática utilizada por preditores de carga é agregar essas características mencionadas ao conjunto de dados, de forma a auxiliar os modelos de previsão a tomar as melhores decisões.

Além destas características, preditores de séries temporais muitas vezes se utilizam de atributos de observação temporal em seus conjuntos de dados, isto é, agregam informações de tempos anteriores para prever o próximo. As técnicas mais comuns de 'adição de características temporais' ou de adição de 'atributos observação temporal', são o *look-back* e as Janelas Deslizantes. Será dedicada uma seção para a apresentação da metodologia destas técnicas.

4.3.1 Atributo 'Tipo de Dia'

Primeiramente foi criado um novo atributo para especificar se a amostra corresponde a um dia útil ou dia não-útil (finais de semana e feriados). Para definição da distinção entre dia útil/não-útil, a própria biblioteca *datetime* do python é capaz de realizar. Para a definição dos demais dias não-uteis, foram considerados os seguintes feriados municipais e nacionais, conforme Rondônia (2021) e Rondônia (2022). Na Tabela 3 são apresentados os feriados considerados.

Para exemplificar esta implementação, é apresentado na Figura 20 o fragmento do conjunto de dados referente ao mês de fevereiro, onde é possível verificar a variação do atributo 'tipo-dia' quando da ocorrência de dias da semana e em finais de semana ou feriados.

4.3.2 Atributos Climáticos

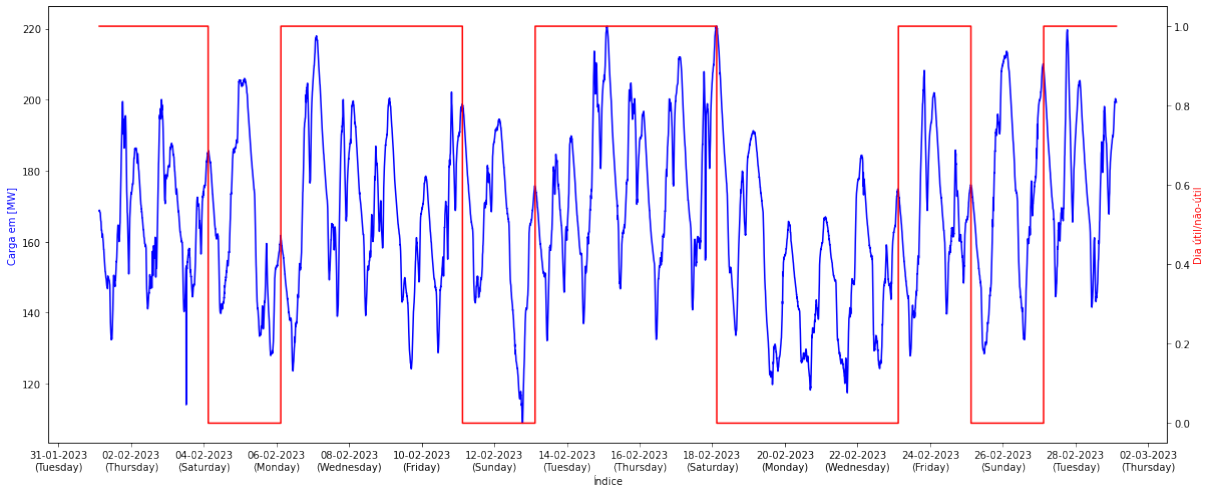
Para a inclusão dos atributos climáticos, foi utilizada a base de dados da Coordenadoria de Recursos Hídricos, vinculada à Secretaria Estadual de Meio Ambiente do estado de Rondônia, disponível em (RONDôNIA, 2023). Essa base de dados foi adotada por estar disponível gratuitamente na internet e por fornecer diversos dados de registros meteorológicos, de hora em hora. Isso facilitou o processo de aquisição de todo os atributos climáticos escolhidos dentro do intervalo de tempo utilizado no conjunto de dados de carregamento dos transformadores.

Tabela 3 – Feriados considerados no conjunto de dados.

Ano	Feriado	Descrição
2022	16/jun	Corpus Christi
2022	07/set	Proclamação da Independência Do Brasil
2022	02/out	Criação do município de Porto Velho
2022	12/out	Padroeira do Brasil Ns[a] Senhora Aparecida
2022	02/nov	Finados
2022	15/nov	Proclamação da República
2022	25/dez	Natal
2023	01/jan	Dia da Confraternização Universal
2023	02/jan	Instalação do estado de Rondônia
2023	23/jan	Instalação do município de Porto Velho
2023	20 - 22/fev	Carnaval
2023	07/abr	Sexta-Feira Santa - Paixão De Cristo
2023	21/abr	Tiradentes

Fonte: Governo do estado de Rondônia

Figura 20 – Exemplo da implementação do atributo 'tipo-dia' no conjunto de dados de Carga no mês de fevereiro



Fonte: do próprio autor.

De todos os atributos disponíveis, foram escolhidos: "Hora Local", "Precipitação", "Umidade Relativa do AR percentual", "Temperatura do ar (°C)" e "Pressão Atmosférica". A Tabela 4 apresenta o *dataset* de atributos climáticos extraídos e posteriormente integrados ao conjunto de dados consolidado contendo a variação da carga no tempo e os tipo de dia:

Tendo organizado os atributos climáticos por data e hora, foi possível sincronizar este *dataframe* com o anterior, formado pelos dados de carga e tipo de dia, por meio da função *merge*, presente na biblioteca do Pandas. Ao integrar os dados climáticos, cuja amostragem é horária, com os dados consolidados de medição da carga, cuja amostragem é de 5 minutos/amostra, estes novos atributos foram repetidos a cada 12 linhas, para que

Tabela 4 – Atributos climáticos integrados ao conjunto de dados.

	Data	**Hora**	**Precip**	**Umid.(%)**	**Temp(C)**	**Pressão Atm.**
0	2022-06-01	0	0.0	97.0	24.5	974.9
1	2022-06-01	1	0.0	98.0	24.4	974.7
2	2022-06-01	2	0.0	92.0	24.9	974.7
3	2022-06-01	3	0.0	93.0	24.7	974.4
4	2022-06-01	4	0.0	95.0	24.4	974.4
...	...	...	...	...	...	...
7950	2023-04-30	19	0.0	82.0	27.9	974.8
7951	2023-04-30	20	0.0	92.0	27.4	975.7
7952	2023-04-30	21	0.0	94.0	27	976
7953	2023-04-30	22	0.0	95.0	26.9	976.5
7954	2023-04-30	23	0.0	88.0	26.8	976.9

Fonte: do próprio autor.

uma hora de medição correspondesse àqueles respectivos dados climáticos. O resultado final, portanto, possui a característica apresentada na Tabela 5:

Tabela 5 – Conjunto de Dados com os atributos climáticos.

	Data	**horario**	**CARGA_PV**	**tipo_dia**	**Precip**	**Umid.(%)**	**Temp(C)**	**P. Atm.**
0	2022-06-03	21:00:00	184.842565	1.0	0.0	96.0	24.6	977.1
1	2022-06-03	21:05:00	184.857166	1.0	0.0	96.0	24.6	977.1
2	2022-06-03	21:10:00	184.808514	1.0	0.0	96.0	24.6	977.1
3	2022-06-03	21:15:00	185.002273	1.0	0.0	96.0	24.6	977.1
4	2022-06-03	21:20:00	185.379640	1.0	0.0	96.0	24.6	977.1
...	...	...	...	...	...	...	...	...
75986	2023-04-30	23:25:00	186.358128	0.0	0.0	88.0	26.8	976.9
75987	2023-04-30	23:30:00	186.639374	0.0	0.0	88.0	26.8	976.9
75988	2023-04-30	23:40:00	140.961629	0.0	0.0	88.0	26.8	976.9
75989	2023-04-30	23:45:00	179.003815	0.0	0.0	88.0	26.8	976.9
75990	2023-04-30	23:50:00	186.606552	0.0	0.0	88.0	26.8	976.9

Fonte: do próprio autor.

4.3.3 Atributos de Observação Temporal Anterior

A análise de séries temporais é desafiadora devido à dependência temporal nos dados. Para superar esse desafio, serão utilizadas duas técnicas semelhantes, mas com algumas diferenças: *look-back* e janelas deslizantes. Ambas as técnicas transformam a série temporal em um problema de aprendizado supervisionado, permitindo que os modelos de aprendizado de máquina padrão possam ser aplicados.

A técnica de *look-back* utiliza informações de etapas de tempo anteriores (*look-back*) para prever a próxima etapa de tempo. Por exemplo, para prever o valor em t+1, podemos usar os valores nos momentos t-2, t-1 e t. Esta técnica pode ser implementada em Python

usando a função '*shift*' do Pandas, que desloca os valores de uma série temporal para frente ou para trás.

Por outro lado, a técnica de janelas deslizantes é semelhante ao *look-back*, mas, em vez de usar um número fixo de observações anteriores, uma "janela"de observações é usada e essa "janela"é "deslizada"através do tempo. Esta técnica pode ser implementada em Python usando a função '*rolling*' do Pandas.

Na Tabela 6 é apresentado o conjunto de dados finalizado, com todos os atributos utilizados adicionados ao conjunto de dados de medição original, inclusive os atributos de observação temporal. Neste exemplo, são apresentados os atributos oriundos da técnica de *look-back*, para um intervalo de t-3.

Tabela 6 – Conjunto de dados definitivo, com todas as *features* adicionadas.

Data	Horário	CARGA	Tipo	Precip	Umid. (%)	Temp. (°C)	Pressão	Lag1	Lag2	Lag3
2022-06-03	21:15:00	185.00	1.0	0.0	96.0	24.6	977.1	184.80	184.85	184.84
2022-06-03	21:20:00	185.37	1.0	0.0	96.0	24.6	977.1	185.00	184.80	184.85
2022-06-03	21:25:00	185.77	1.0	0.0	96.0	24.6	977.1	185.37	185.00	184.80
2022-06-03	21:30:00	185.80	1.0	0.0	96.0	24.6	977.1	185.77	185.37	185.00
2022-06-03	21:35:00	185.83	1.0	0.0	96.0	24.6	977.1	185.80	185.77	185.37
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
2023-04-30	23:25:00	186.35	0.0	0.0	88.0	26.8	976.9	185.93	185.84	185.86
2023-04-30	23:30:00	186.63	0.0	0.0	88.0	26.8	976.9	186.35	185.93	185.84
2023-04-30	23:40:00	140.96	0.0	0.0	88.0	26.8	976.9	186.63	186.35	185.93
2023-04-30	23:45:00	179.00	0.0	0.0	88.0	26.8	976.9	140.96	186.63	186.35
2023-04-30	23:50:00	186.60	0.0	0.0	88.0	26.8	976.9	179.00	140.96	186.63

Fonte: do próprio autor.

Ao implementar essas técnicas nos modelos adotados, buscou-se uma melhoria significativa no desempenho de cada um deles. Para os modelos baseados em técnicas tradicionais e os baseados em árvore de decisão, foi empregada a técnica de *look-back*, enquanto para as redes neurais, foram utilizadas janelas deslizantes.

No próximo capítulo, portanto, serão apresentados exemplos dos processos de previsão com e sem a utilização do *look-back*/janela deslizantes. Isso permitirá uma visualização mais clara do impacto desta técnica na qualidade das previsões, tanto visualmente quanto por meio de métricas de avaliação.

4.4 Metodologia da Análise Exploratória de Dados (AED)

A análise exploratória de dados (AED) é uma etapa essencial no processo de análise de dados, que ocorre antes da aplicação de técnicas de modelagem mais formais. A AED permite a uma percepção valiosa e aprofundada do conjunto de dados, trazendo à tona as características essenciais, os padrões ocultos, as relações subjacentes, os *outliers* e as características intrínsecas que podem não ser visíveis com a aplicação direta de modelos

preditivos. O principal objetivo da AED é maximizar a visão do analista sobre um conjunto de dados e sobre a estrutura subjacente dos dados.

A AED subsidia a tomada de decisões informadas sobre quais variáveis têm o maior impacto ou relação com a variável alvo, quais são os possíveis problemas de multicolinearidade, onde os *outliers* podem estar afetando os dados, e como as variáveis interagem entre si. A AED também pode orientar as decisões sobre quais transformações de dados podem ser necessárias para melhorar a qualidade dos dados e o desempenho dos modelos, como a normalização ou a padronização.

No caso específico de tratamento de séries temporais, a AED desempenha um papel crucial, fornecendo informações sobre a tendência, a sazonalidade e a autocorrelação dos dados. Isto é fundamental para a escolha dos modelos de previsão apropriados e para a validação das suposições de modelagem.

Em resumo, a Análise Exploratória de Dados não é apenas um passo preliminar, mas um componente integral do processo de análise de dados. Ela oferece o suporte necessário para a seleção do modelo apropriado, melhora a precisão das previsões e fornece um entendimento mais profundo da natureza dos dados em mãos.

Os resultados das análises exploratórias do conjunto de dados final será apresentado no próximo capítulo deste trabalho.

4.4.1 Análise Descritiva

Antes de iniciar a modelagem preditiva, é útil obter uma descrição resumida dos dados, conhecida como análise descritiva. Isto pode incluir medidas de tendência central como a média e a mediana, medidas de dispersão como o desvio padrão e o intervalo interquartil, e a distribuição dos dados.

Em resumo, a média nos dá uma medida de centralidade dos dados; a mediana é uma medida robusta de centralidade que é menos suscetível a *outliers*; o desvio padrão nos informa sobre a dispersão dos dados ao redor da média; enquanto a curtose e a assimetria fornecem informações sobre a forma da distribuição dos dados.

A análise de correlação, por sua vez, pode ser extremamente útil para entender as relações entre os diferentes atributos, medindo a relação linear entre dois atributos. Seu resultado varia de -1 a 1, onde -1 indica uma relação linear negativa perfeita (à medida que um atributo aumenta, o outro diminui), 1 indica uma relação linear positiva perfeita (à medida que um atributo aumenta, o outro também aumenta), e 0 indica que não há relação linear entre os atributos.

A biblioteca pandas do Python é uma ferramenta extremamente poderosa para realizar análise estatística descritiva. Com o método *'describe()'* em um *dataframe* pandas, pode-se obter um resumo estatístico dos dados que inclui várias das métricas mencionadas.

Já a correlação, pode ser calculada no Python usando a função *'.corr()'*.

4.4.2 Análise Gráfica

A Análise Gráfica é uma importante etapa na Análise Exploratória de Dados que permite visualizar as distribuições de dados, padrões, anomalias e relações entre as variáveis. É fundamental na tomada de decisão para a escolha de métodos de modelagem mais adequados ao tipo de dados em questão.

Geralmente, as análises gráficas começam com gráficos univariados que permitem entender a distribuição, a centralidade e a dispersão de cada variável individualmente. Por exemplo, os histogramas e os *boxplots* nos ajudam a visualizar a distribuição de uma variável, identificar a presença de *outliers* e compreender a simetria dos dados.

Em seguida, as análises gráficas bivariadas e multivariadas são úteis para explorar as relações entre duas ou mais variáveis. Por exemplo, os gráficos de dispersão (*scatterplots*) permitem verificar se existe alguma relação linear ou não-linear entre as variáveis.

Outros tipos de gráficos, como gráficos de linha, são particularmente úteis ao lidar com dados temporais, pois nos permitem visualizar tendências, sazonalidades e outras características que só podem ser observadas ao longo do tempo.

A seguir, serão apresentados as ferramentas de análise gráficas adotadas neste trabalho com uma breve descrição:

Histogramas: Os histogramas são uma das primeiras ferramentas mais comuns utilizadas na análise exploratória. Eles representam a distribuição de um único atributo, dividindo-o em uma série de bins e contando o número de observações que caem em cada bin. A forma do histograma pode fornecer dicas valiosas sobre a distribuição dos dados (Gaussiana, Exponencial, Bimodal, etc.). Em Python, os histogramas podem ser criados com a função *hist()* do Matplotlib ou a função *distplot()* do Seaborn.

Boxplots: Os *boxplots* são uma ferramenta padrão na análise exploratória de dados que fornece uma representação visual da posição central, dispersão e, mais importante, possíveis *outliers*. Ele divide os dados em quartis e representa a variabilidade de dados além do alcance interquartil (IQR), que são os valores dentro dos 25% e 75% dos dados. *Boxplots* podem ser criados com a função *boxplot()* do Seaborn.

Gráficos de Dispersão: Gráficos de dispersão são utilizados para visualizar a relação entre dois atributos. Cada ponto no gráfico de dispersão representa uma observação no conjunto de dados. Os gráficos de dispersão são particularmente úteis para identificar tendências, correlações e padrões em pares de atributos. No Seaborn, gráficos de dispersão podem ser criados usando a função *scatterplot()*.

Gráficos de Linha: Gráficos de linha são úteis para visualizar tendências temporais ou sequenciais em um conjunto de dados. No nosso caso, um gráfico de linha das leituras

de carga ao longo do tempo pode revelar tendências diárias, semanais ou sazonais. O Matplotlib fornece a função *plot()* para criar gráficos de linha.

Gráficos de Correlação: Também conhecidos como gráficos de matriz de correlação, são gráficos que apresentam a correlação entre várias variáveis em um único gráfico em forma de matriz. Cada célula na matriz é colorida de acordo com o valor de correlação entre as variáveis correspondentes. Isso pode ser feito no Python usando a função *heatmap()* do Seaborn.

4.4.3 Testes de Estacionariedade

Na análise de séries temporais, uma premissa fundamental para muitos dos métodos de modelagem é que a série em análise seja estacionária. Uma série estacionária é aquela cujas propriedades estatísticas, como média, variância e autocorrelação, são constantes ao longo do tempo. Essa estabilidade ao longo do tempo permite que os modelos façam previsões futuras que são confiáveis.

O Teste de Dickey-Fuller Aumentado (*Augmented* Dickey-Fuller - ADF) é um teste estatístico comum usado para testar a hipótese nula de que uma série temporal possui uma raiz unitária contra a alternativa de estacionariedade (ou seja, a série é estacionária). O teste utiliza uma abordagem de regressão autoregressiva e considera a estrutura de autocorrelação nos dados ao testar a hipótese.

Se o teste ADF rejeitar a hipótese nula, isso significa que a série temporal é estacionária e, portanto, adequada para a modelagem. Se a hipótese nula não for rejeitada, isso implica que a série temporal não é estacionária, sugerindo a necessidade de pré-processamento adicional ou a adoção de um modelo de séries temporais mais sofisticado que possa lidar com a não estacionariedade.

Para realizar este teste em Python, foi utilizada a biblioteca *statsmodels* ferramenta *adfuller*.

4.4.4 Testes de Autocorrelação

A autocorrelação é uma característica importante das séries temporais. É uma estatística que mede a relação linear entre observações em *lag*, isto é, em atraso uma em relação a outra em uma série de tempo. Em termos simples, a autocorrelação verifica como uma série de tempo está relacionada a si mesma em diferentes *lags* de tempo. Por exemplo, em uma série temporal carga elétrica, pode haver autocorrelação positiva entre a carga em um dia e no dia, se o perfil de consumo for regular.

Os testes de autocorrelação realizados neste trabalho foram:

Função de Autocorrelação (ACF): Este é um gráfico que mostra a autocorrelação de uma série temporal em diferentes *lags*. O eixo x representa o *lag* e o eixo y

representa a força da autocorrelação. Barras que se estendem além da linha pontilhada indicam autocorrelação significativa nesse *lag*.

Função de Autocorrelação Parcial (PACF): Semelhante à ACF, mas mostra a autocorrelação em um *lag*, controlando os valores em *lags* menores. É útil para identificar o possível *lag* em um modelo de Autoregressão (AR).

Gráfico de Latência ou *Lag plot*: O gráfico de latência é uma ferramenta gráfica simples que verifica a autocorrelação plotando a observação no tempo t no eixo x e a observação no tempo t+1 no eixo y. Se os dados são altamente autocorrelacionados, os pontos no *Lag plot* serão aproximadamente dispostos ao longo de uma linha diagonal.

Teste de Durbin-Watson (DW): Este é um teste estatístico que é comumente usado para detectar a presença de autocorrelação nos resíduos de uma análise de regressão. Um valor DW perto de 2 sugere que não há autocorrelação, enquanto um valor DW significativamente menor que 2 indica autocorrelação positiva e um valor DW significativamente maior que 2 indica autocorrelação negativa.

Teste de Ljung-Box: Este é um tipo de teste estatístico que testa a hipótese nula de que os dados são independentemente distribuídos. Em outras palavras, testa a hipótese nula de que não há autocorrelação em um conjunto de dados de séries temporais.

Entender a autocorrelação em uma série temporal é importante por várias razões. Primeiro, a presença de autocorrelação pode violar os pressupostos de independência em modelos de regressão clássica, levando a estimativas de parâmetros tendenciosas e a subestimação da incerteza. Segundo, a autocorrelação pode ser uma característica importante e informativa de uma série temporal, e modelos como AR, MA e ARIMA usam essa propriedade para fazer previsões.

4.4.5 Testes de Normalidade

Os testes de normalidade são usados para determinar se um conjunto de dados se aproxima de uma distribuição normal. Eles são importantes porque muitos dos testes estatísticos adotados assumem que que os dados são normalmente distribuídos. Se essas suposições não forem cumpridas, os testes estatísticos podem ser inválidos.

Os dois testes de normalidade mais comuns são o teste de Shapiro-Wilk e o teste de Anderson-Darling.

Teste de Shapiro-Wilk: É uma maneira de testar a normalidade. Este teste produz uma estatística, W, e um valor-p. Se o valor-p for menor do que o nível de significância pré-especificado (geralmente 0,05), então a hipótese nula de normalidade é rejeitada e existe evidência para afirmar que os dados testados não são normalmente distribuídos.

Teste de Anderson-Darling: Outro teste de normalidade. O teste de Anderson-

Darling retorna uma estatística 'A' e um conjunto de valores críticos. Se a estatística 'A' for maior do que o valor crítico para um determinado nível de significância, então a hipótese nula de normalidade é rejeitada.

Os testes de Shapiro-Wilk e Anderson-Darling de podem ser utilizados através da biblioteca *scipy.stats* do Python.

A interpretação dos resultados de todos esses testes e análises ajudam a construir a melhor abordagem de modelagem para, neste trabalho, prever a carga elétrica.

4.5 Aplicação dos Modelos de Predição

Na continuidade da análise, a exploração dos dados foi aprofundada, concentrando na implementação de uma série de técnicas avançadas de modelagem para a previsão da carga nos transformadores de potência, objeto desse estudo. Para esta fase do estudo, foram aplicados as seguintes técnicas:

- Regressão Linear e Polinominal;
- *Support Vector Machine* (SVM);
- Modelos baseados em Árvore de Decisão;
  - *Random Forest*
  - *XGBoost*
- Redes Neurais
  - Rede Neural Artificial - ANN
  - Rede Neural Convolucional - CNN
  - *Long Short Term Memory* - LSTM
  - *Gated Recurrent Units* - GRU

Em comum, todos os modelos foram submetidos ao mesmo subconjunto de treinamento e teste[2]. Isto significa que todos os modelos foram expostos à mesma quantidade e qualidade de dados durante o treinamento e foram testados contra o mesmo conjunto de dados desconhecidos. Esta consistência na preparação e utilização dos dados garante que as diferenças de desempenho observadas entre os modelos sejam verdadeiramente representativas de suas capacidades relativas de aprendizado e previsão, em vez de serem influenciadas pela quantidade ou qualidade dos dados de treinamento ou teste.

[2] Os modelos baseados em redes neurais foram treinados e testados sendo submetidos ao conjunto de dados contendo apenas os dados de carga, isto é, sem atributos climáticos ou de característica diária.

Quanto à divisão entre os dois conjuntos, 80% dos dados foram usados para treinamento e os 20% restantes para teste. Esta proporção foi escolhida para garantir que os modelos tivessem uma quantidade substancial de dados para aprender, ao mesmo tempo que reservava uma quantidade suficiente de dados para testar e validar a precisão e a generalização dos modelos.

Na Tabela 7 é apresentado o intervalo utilizado na separação do conjunto de treinamento e teste, utilizado por todos os modelos neste trabalho.

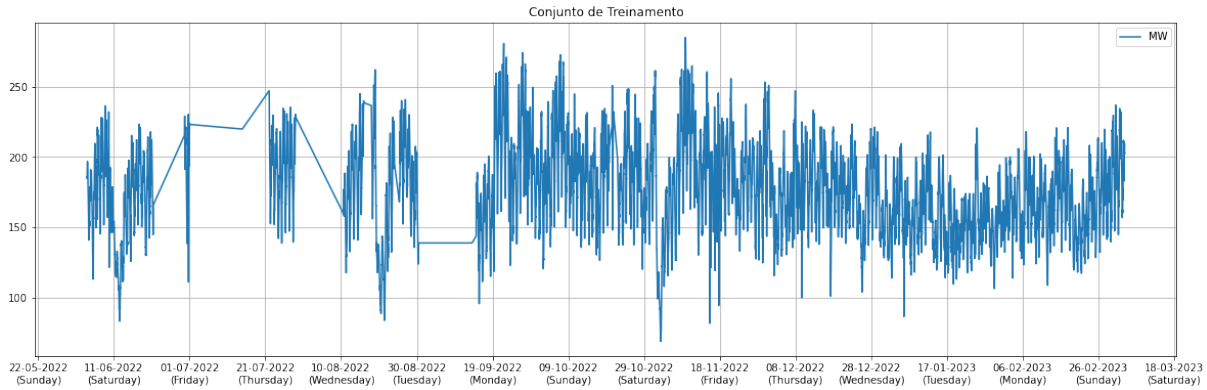
Tabela 7 – Conjunto de Treinamento e Teste

data_horario	y_train	data_horario	y_test
2022-06-03 21:00:00	184.842565	2023-03-05 00:00:00	202.299553
2022-06-03 21:05:00	184.857166	2023-03-05 00:05:00	202.173123
2022-06-03 21:10:00	184.808514	2023-03-05 00:10:00	202.023641
2022-06-03 21:15:00	185.002273	2023-03-05 00:15:00	201.961467
2022-06-03 21:20:00	185.379640	2023-03-05 00:20:00	203.326187
...	...	...	...
2023-03-04 23:35:00	202.232101	2023-04-30 23:25:00	186.358128
2023-03-04 23:40:00	202.464983	2023-04-30 23:30:00	186.639374
2023-03-04 23:45:00	202.351465	2023-04-30 23:40:00	140.961629
2023-03-04 23:50:00	202.370473	2023-04-30 23:45:00	179.003815

Fonte: do próprio autor.

O intervalo de cada conjunto é apresentado graficamente nas Figuras 21 e 22.

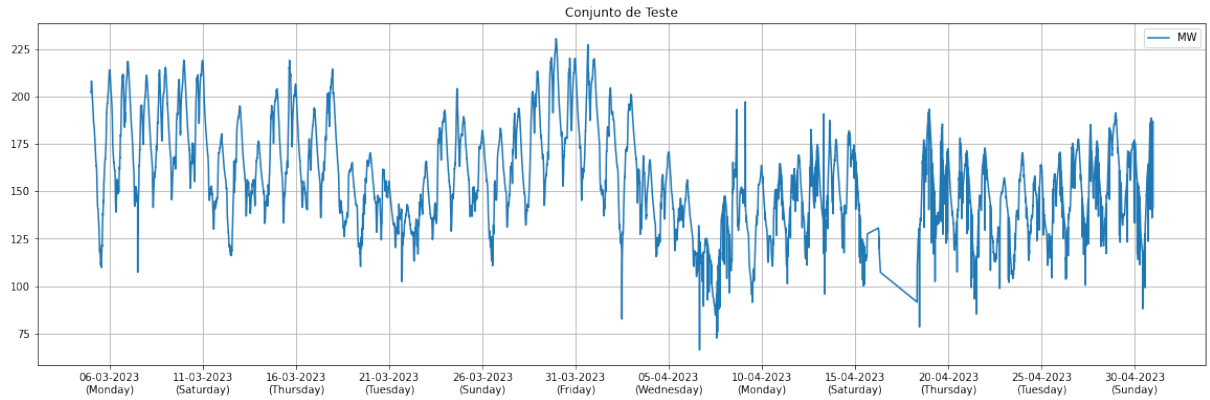
Figura 21 – Conjunto de Treinamento



Fonte: do próprio autor.

Adicionalmente, foi realizada a normalização dos dados de entrada. Este foi um passo crítico no processo de preparação dos dados para muitos dos modelos de previsão. A normalização dos dados permite que os algoritmos operem sob a suposição de que os dados de entrada estão na mesma escala, removendo assim quaisquer vieses que possam ser introduzidos por diferenças nas magnitudes, unidades ou escalas dos atributos dos dados. Isto é especialmente importante para algoritmos que dependem da medição de

Figura 22 – Conjunto de Teste



Fonte: do próprio autor.

distâncias entre pontos de dados, como o *Support Vector Machine* (SVM) e a regressão linear e polinomial, bem como para algoritmos que usam gradientes para encontrar o mínimo, como as Redes Neurais Artificiais (ANN), Redes Neurais Convolucionais (CNN), *Long Short-Term Memory* (LSTM) e *Gated Recurrent Units* (GRU).

Neste trabalho, foi utilizado o método *MinMaxScaler()* para normalizar os dados. Este método transforma os dados de modo que todos os valores caem no intervalo de 0 a 1. O MinMaxScaler faz isso subtraindo o valor mínimo do conjunto de dados de cada ponto de dados e então dividindo pelo intervalo dos dados (o valor máximo menos o valor mínimo). Este método é uma escolha comum para a normalização de dados por sua simplicidade e eficácia, e tem a vantagem de preservar a forma da distribuição original dos dados, não alterando a informação contida nos valores relativos dos pontos de dados.

Além disso, novamente, para garantir a consistência na comparação do desempenho dos diferentes modelos, foi aplicado o mesmo método de normalização a todos os conjuntos de dados antes do treinamento dos modelos. Isso garante que todos os modelos recebam dados de entrada que tenham sido preparados e transformados da mesma maneira.

Com relação aos modelos utilizados, toda a fundamentação teórica de cada modelo foi apresentada no capítulo anterior. Aqui serão apresentadas apenas aspectos de sua modelagem, como a arquitetura utilizada em cada modelo.

4.5.1 Regressão Linear e Polinominal

O modelo de Regressão Linear utilizado no trabalho está disponível no pacote de aprendizado de máquina Scikit-learn, que fornece a classe *LinearRegression*, implementando a regressão linear com o método de mínimos quadrados.

A diferença para o modelo de regressão polinomial é que é implementada adicionando a transformação *PolynomialFeatures* num processo anterior ao treinamento do modelo. Neste trabalho, foi utilizado o modelo de grau 3. Não obstante, o modelo resultante é ainda

um modelo linear porque a regressão é linear nos parâmetros, e os termos polinomiais são apenas termos de entrada transformados (SCIKITLEARN, 2023b).

4.5.2 *Support Vector Machine* - SVM

O *Suport Vector Machine* (SVM) é um algoritmo de aprendizado de máquina supervisionado que pode ser usado tanto para classificação como para regressão. Ele é popular por sua capacidade de lidar com dimensões de alta ordem e sua eficácia em espaços de grandes dimensões. O modelo Python utilizado neste trabalho, através da biblioteca Scikit-learn foi o módulo SVR (SCIKITLEARN, 2023c).

Foram utilizadas duas variantes do SVM neste trabalho: uma com o *kernel Radial Basis Function (RBF)* e a outra com o *kernel* polinomial. A descrição de ambas foram apresentados na Tabela 1 no Capítulo 2.

A Tabela 8 apresenta os parâmetros utilizados no *kernel* de cada modelo. A determinação destes parâmetros foi definida por processo iterativo, até que fosse encontrado os melhores resultados de generalização.

Tabela 8 – Parâmetros utilizados nos modelos SVM

	C	gamma	epsilon	degree	coef
kernel RBF	150	0.002	0.1		
kernel poly	100	scale	0.1	3	1

Fonte: do próprio autor.

4.5.3 Modelos Baseados em Árvore de Decisão

Neste trabalho, foram adotados os modelos *Random Forest* e *XGBoost*.

O *Random Forest* é um método de aprendizado de conjunto que opera através da construção de uma infinidade de árvores de decisão no momento do treinamento e gerando a classe que é o modo das classes ou a média de previsão das árvores individuais. No Python, a biblioteca Scikit-learn fornece uma implementação eficiente do algoritmo *Random Forest* através do módulo *RandomForestRegressor* (SCIKITLEARN, 2023a).

No contexto deste trabalho, utilizamos p parâmetro 'n_estimators=100', o que significa foram criados 100 árvores de decisão para realizar a previsão dos dados de carga.

Por outro lado, o *XGBoost* é uma implementação avançada e eficiente do algoritmo de *Gradient Boosting*. Diferentemente do *Random Forest*, que constrói e combina árvores de forma independente do desempenho das outras, o *XGBoost* constrói uma árvore de cada vez, onde cada nova árvore ajuda a corrigir os erros cometidos pela árvore anteriormente treinada (XGBOOST, 2023).

A biblioteca *XGBoost* é inerente ao python e pode ser utilizado através de comando de *import*, desde que previamente instalado.

Os parâmetros utilizados no modelo *XGBoost* utilizado neste trabalho estão exibidos na Tabela 9

Tabela 9 – Parâmetros utilizados no modelo *XGBoost*

objective	colsample_bytree	learning_rate	max_depth	alpha	n_estimators
reg:squarederror	1.0	0.25	5	10	200

Fonte: do próprio autor.

4.5.4 Redes Neurais

As redes neurais são amplamente utilizadas em problemas de aprendizado de máquina e, neste trabalho, foram exploradas quatro arquiteturas: ANN, CNN, GRU e LSTM.

A biblioteca Keras, uma API de alto nível para construir e treinar modelos de aprendizado profundo, foi utilizada para implementar todas as arquiteturas de redes neurais mencionadas (KERAS, 2023). Keras é construído em cima do TensorFlow (TENSORFLOW, 2023), uma poderosa biblioteca de computação numérica desenvolvida pela Google Brain, que fornece as bases para a construção e o treinamento de modelos de aprendizado profundo. A arquitetura de cada modelo foi configurada usando a classe Sequential do Keras, que permite a construção de um modelo camada por camada.

A Rede Neural Artificial (ANN) usada consiste em uma camada de entrada com 64 neurônios e função de ativação *'sigmoid'*, duas camadas de *dropout* para regularização (ajudando a evitar o overfitting), uma camada oculta com 32 neurônios e função de ativação *'sigmoid'*, e uma camada de saída com 1 neurônio.

A Rede Neural Convolucional (CNN) utilizada tem uma camada convolucional de 64 filtros e tamanho de *kernel* 3, seguida por uma camada de *pooling* para reduzir a dimensionalidade, uma camada *'Flatten'* para achatamento dos dados, uma camada densa com 50 neurônios e função de ativação 'relu', e uma camada de saída com 1 neurônio.

A Rede Neural Recorrente GRU usada tem uma camada GRU com 64 unidades seguida por uma camada de saída com 1 neurônio.

A Rede Neural LSTM (*Long Short-Term Memory*) utilizada tem uma camada LSTM de 64 unidades, duas camadas de dropout, uma segunda camada LSTM de 32 unidades, e uma camada de saída com 1 neurônio.

A Tabela 10 descreve as arquiteturas usadas:

Novamente, para efeito de comparação, todos os modelos foram compilados com o otimizador 'adam' e a função de perda 'mse' (erro quadrático médio).

Tabela 10 – Arquiteturas das Redes Neurais utilizadas

Modelo de Rede Neural	**Arquitetura**	**Funções de Ativação**
ANN	1. Camada densa com 64 neurônios 2. Dropout (0.2) 3. Camada densa com 32 neurônios 4. Dropout (0.2) 5. Camada densa com 1 neurônio	1. Sigmoid 3. Sigmoid
CNN	1. Camada convolucional 1D com 64 filtros 2. Max Pooling 1D (tamanho do pool: 2) 3. Flatten 4. Camada densa com 50 neurônios 5. Camada densa com 1 neurônio	1. ReLU 4. ReLU
GRU	1. Camada GRU com 64 unidades 2. Camada densa com 1 neurônio	1. Tanh
LSTM	1. Camada LSTM com 64 unidades (retorno desequências: True) 2. Dropout (0.2) 3. Camada LSTM com 32 unidades 4. Dropout (0.2) 5. Camada densa com 1 neurônio	1. Tanh 3. Tanh

Fonte: do próprio autor.

Todas as redes neurais foram treinadas e testadas considerando o conjunto de dados como uma série temporal univariável, isto é, apenas uma variável foi considerada ao longo do tempo, a "Carga_PV", sem a inclusão de múltiplas características ou variáveis independentes, como é o caso dos atributos climáticos ou do tipo de dia. Isso significa que as previsões foram baseadas exclusivamente no histórico da variável alvo, e outras informações potencialmente correlacionadas ou influentes foram excluídas do modelo. Testes com o conjunto de dados com todos os atributos mostraram um aumento do custo computacional sem grandes - ou até mesmo nenhum - ganho de qualidade no resultado.

4.6 Avaliação da Influência das *Features*

Na análise estatística descritiva, abordada em 4.4.1, foi mencionada a utilização do teste de correlação entre as *features* para avaliar a influência de cada característica na nossa variável alvo. A correlação é uma medida que varia de -1 a 1 e que indica a força e a direção do relacionamento linear entre duas variáveis. Um valor próximo de 1 indica uma forte correlação positiva, enquanto um valor próximo de -1 indica uma forte correlação negativa. Complementando essa análise, também realizamos análises gráficas que serão apresentadas em detalhes no capítulo de resultados. A análise de correlação já é um bom indicativo para mensurar a influência das características adicionadas ao *dataframe* sobre a

variável de interesse 'carga'.

Contudo, para uma avaliação mais robusta da importância das características, foi utilizada a função 'OLS' (*Ordinary Least Squares*), da biblioteca statsmodels. Esta função é um método que busca encontrar a melhor linha de regressão linear que se ajusta aos dados. Esta técnica envolve o cálculo dos coeficientes que minimizam a soma dos quadrados das diferenças entre os valores observados e previstos.

Na saída da análise OLS, vários parâmetros são fornecidos para ajudar a avaliar o ajuste e a importância de cada característica, entre eles:

- **R-squared:** Este é o coeficiente de determinação, que representa a proporção da variação da variável dependente que é previsível pelas variáveis independentes. Um valor próximo a 1 indica um ajuste perfeito do modelo.
- **Adj. R-squared:** Similar ao R-squared, mas ajustado pelo número de preditores no modelo. Ajuda a evitar a interpretação errônea de um bom ajuste quando variáveis adicionais são incluídas no modelo.
- **coef:** Estes são os coeficientes da regressão para cada variável, representando o quanto a variável de resposta muda quando essa característica muda em uma unidade.
- **std err:** O erro padrão dos coeficientes, que mede o nível de incerteza associado aos coeficientes.
- **t**: O valor-t, que é usado para testar a hipótese nula de que o coeficiente é igual a zero (ou seja, não tem efeito).
- **P>|t|:** O valor-p associado ao valor-t, que ajuda a determinar a significância estatística dos coeficientes.
- **[0.025 e 0.975]:** Estes são os limites do intervalo de confiança de 95% para os coeficientes. Se esse intervalo não incluir zero, é um indicador de que a característica é estatisticamente significativa.

Os coeficientes representam a sensibilidade da variável de resposta em relação aos preditores e são essenciais para entender o impacto de cada característica na variável alvo. Por exemplo, um coeficiente positivo indica que, à medida que o valor da característica aumenta, a variável de resposta também aumenta, mantendo-se constante as demais variáveis.

Além da análise de correlação e da regressão linear OLS, outra abordagem utilizada para avaliar a importância das características foi o recurso *'FeatureImportances'* (SHIN, 2021). Esta é uma técnica que avalia a importância de cada característica com base no

quanto cada característica contribui para aumentar a precisão do modelo. Por exemplo, nos modelos de árvores de decisão, como o *Random Forest* e o *XGBoost*, a importância de uma característica pode ser avaliada com base em quão frequentemente a característica é utilizada para dividir os dados. Isso fornece uma medida quantitativa da influência de cada característica na performance do modelo.

4.7 Metodologia da Avaliação do Desempenho dos Modelos

A avaliação do desempenho dos modelos de aprendizado de máquina é um passo crucial na modelagem preditiva. Medir a eficácia do modelo nos ajuda a entender até que ponto podemos confiar nas previsões fornecidas pelo nosso modelo, além de ser uma maneira de comparar diferentes modelos e escolher o que melhor se ajusta aos nossos dados.

4.7.1 Métricas de Desempenho

Existem diversas métricas que podem ser usadas para avaliar a performance de um modelo de aprendizado de máquina. A escolha da métrica depende do tipo de problema de aprendizado de máquina que estamos enfrentando (classificação, regressão, clusterização, etc.), da natureza dos dados e das exigências específicas do problema que estamos tentando resolver.

Para problemas de regressão, as métricas de desempenho comuns incluem: Erro Médio Absoluto (Mean Absolute Error - MAE), Erro Quadrático Médio (Mean Squared Error - MSE), Raiz do Erro Quadrático Médio (Root Mean Squared Error - RMSE), Coeficiente de Determinação (R^2), entre outras.

Neste trabalho, foram utilizados o MSE e o R^2 para avaliar o desempenho dos nossos modelos.

O MSE é uma métrica que mede a média dos erros quadráticos, onde o erro é a diferença entre o valor predito pelo modelo e o valor real. Ele é calculado pela seguinte fórmula:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2, \tag{4.1}$$

onde y_i é o valor real do i-ésimo ponto no conjunto de dados, $\hat{y}_i$ é o valor predito pelo modelo para o i-ésimo ponto no conjunto de dados, e n é o número total de pontos no conjunto de dados.

O MSE é uma métrica útil pois penaliza mais erros grandes do que pequenos, sendo, portanto, sensível a *outliers*. É uma métrica que mede a média dos erros quadráticos, penalizando mais erros grandes do que pequenos. Portanto, se obtivermos um MSE muito

grande, isso indica que nosso modelo está cometendo erros grandes em suas previsões. Em outras palavras, a diferença entre os valores reais e os valores preditos pelo modelo é grande. Isso é um indicativo de que o modelo pode não ser adequado para os dados ou que pode estar subajustado ou sobreajustado.

O coeficiente de determinação, ou R^2, é uma métrica que indica a proporção da variância nos dados que é prevista pelo modelo. É calculado pela seguinte fórmula:

$$R^2 = 1 - \frac{SSR}{SST}, \tag{4.2}$$

onde SSR é a soma dos quadrados residuais, que é a soma das diferenças ao quadrado entre os valores preditos e os reais, e SST é a soma dos quadrados totais, que é a soma das diferenças ao quadrado entre os valores reais e a média dos valores reais. R^2 varia entre 0 e 1, onde um valor de 1 indica que o modelo é capaz de prever perfeitamente a variável dependente usando as variáveis independentes, e um valor de 0 indica que o modelo não é capaz de prever a variável dependente usando as variáveis independentes. No entanto, em alguns casos, podemos obter um R^2 negativo. Isso geralmente acontece quando o modelo é tão ruim que a linha de melhor ajuste do modelo é pior do que uma linha horizontal simples traçada através da média dos dados. Ou seja, o modelo é pior do que um modelo que faz previsões simplesmente baseadas na média dos valores reais. Isso também é um indicativo de que o modelo pode não ser adequado para os dados ou que pode estar subajustado ou sobreajustado.

Essas duas métricas foram escolhidas porque juntas oferecem uma visão completa da performance do modelo: o MSE fornece uma quantificação do erro de previsão, enquanto o R^2 fornece uma medida de quão bem o modelo pode capturar a variabilidade nos dados. O MSE, sendo uma métrica absoluta, permite avaliar a magnitude dos erros do modelo, enquanto o R^2, uma métrica relativa, permite avaliar a proporção da variância nos dados que o modelo é capaz de explicar. Essa combinação fornece uma visão abrangente do desempenho do modelo, permitindo entender não apenas o tamanho dos erros, mas também a capacidade do modelo de capturar a estrutura subjacente dos dados. Deste modo, apesar da existência de diversas métricas de avaliação, optou-se por utilizar o MSE e o R^2 neste trabalho por serem suficientes para fornecer uma avaliação robusta e compreensível da eficácia dos modelos utilizados.

Além das métricas de erro e de ajuste do modelo, o tempo de processamento também foi considerado na avaliação do desempenho dos modelos. Entende-se que, para aplicações práticas, especialmente em tempo real ou quase real, um bom modelo não deve apenas ter bom desempenho em termos de precisão, mas também deve ser eficiente em termos de tempo de processamento. Para medir o tempo de execução tanto para o treinamento quanto para a fase de teste dos modelos, foi utilizado o comando mágico

'%%time' do IPython. Este comando, quando colocado no início de uma célula de código, fornece informações sobre o tempo de execução da célula, incluindo o tempo de processador (CPU time), o tempo de execução em tempo real (Wall time) e outros. Todos os resultados foram gerados utilizando o mesmo equipamento, cuja especificação está descrita da Tabela 11.

Tabela 11 – Especificações do Sistema

Componente	**Especificação**
Processador	Intel(R) Core(TM) i7-6700 CPU @ 3.40GHz 3.41 GHz
RAM instalada	32.0 GB
Tipo de sistema	Sistema operacional de 64 bits, processador baseado em x64
Sistema Operacional	Windows 10 Pro

Fonte: do próprio autor.

A avaliação dos modelos neste trabalho, portanto, não se baseou apenas na qualidade da previsão, mas também na eficiência computacional. Os melhores modelos serão identificados como aqueles que fornecem previsões de alta qualidade (baixo MSE, R^2 próximo de 1) e que são eficientes em termos de tempo de processamento. Desta forma, foi possível comparar os modelos não apenas em termos de seu desempenho de previsão, mas também em termos de sua adequação para uso em aplicações em tempo real ou quase real, onde a eficiência do tempo de processamento é crucial.

4.7.2 Análise Gráfica dos Resultados de Previsão

Na última seção deste capítulo, abordamos a 'Análise Gráfica dos Resultados de Previsão', que é um componente vital da avaliação de qualquer modelo de aprendizado de máquina. Essa análise permite uma compreensão visual intuitiva do desempenho do modelo, permitindo a comparação entre os resultados preditos pelo modelo e o conjunto de dados de teste. Todos os modelos analisados neste trabalho foram treinados com o mesmo conjunto de dados de treinamento para garantir uma comparação justa e direta entre eles.

Em Python, a biblioteca 'matplotlib' é comumente usada para traçar gráficos, juntamente com a 'seaborn' para um visual mais sofisticado. Essas bibliotecas fornecem uma variedade de ferramentas gráficas que permitem uma análise abrangente e visualmente intuitiva do desempenho do modelo.

A avaliação gráfica dos resultados preditos começa com o plot dos resultados reais do conjunto de teste em contraste com as previsões do modelo. Uma boa aderência entre as curvas representando os dados de teste e as previsões indica um modelo que generaliza bem e pode ser considerado confiável.

Além disso, serão apresentadas as curvas de aprendizado dos modelos. A curva de aprendizado mostra a evolução do desempenho do modelo durante o treinamento e é

uma excelente ferramenta para avaliar se o modelo está aprendendo corretamente. Para os modelos que não apresentam um método explícito para plotar a curva de aprendizado, será apresentada a curva de perda (Loss), que representa a diferença entre as previsões do modelo e os verdadeiros valores alvo durante o treinamento.

A curva de aprendizado e a curva de perda são ferramentas essenciais para entender o comportamento de um modelo durante o treinamento. A curva de aprendizado registra o desempenho do modelo no conjunto de treinamento e validação à medida que a quantidade de dados de treinamento aumenta. Já a curva de perda exibe a magnitude da função de perda ao longo das épocas de treinamento.

Uma curva de aprendizado ideal apresenta uma tendência de diminuição do erro de treinamento à medida que mais dados são incorporados ao treinamento, até chegar a um ponto em que adicionar mais dados não resulta em melhorias significativas no desempenho. Adicionalmente, o desempenho no conjunto de validação deve começar relativamente alto, mas rapidamente alcançar e acompanhar a tendência de diminuição do erro de treinamento. A convergência dessas duas curvas indica que o modelo está generalizando bem e não está sobreajustando aos dados de treinamento.

No caso da curva de perda, um bom resultado é indicado por uma tendência de queda consistente à medida que as épocas de treinamento progridem, idealmente se estabilizando em um valor baixo. Caso a curva de perda aumente ou flutue drasticamente, isso pode indicar problemas como sobreajuste, onde o modelo está se adaptando demais aos dados de treinamento e perdendo a capacidade de generalizar para novos dados, ou subajuste, onde o modelo não é complexo o suficiente para capturar a estrutura subjacente dos dados.

Portanto, a análise das curvas de aprendizado e perda fornece *insights* valiosos sobre a eficácia do processo de treinamento, o desempenho do modelo e possíveis ajustes que podem ser feitos para melhorar o resultado final. Essas análises são fundamentais para alcançar um equilíbrio entre o ajuste e a capacidade de generalização dos modelos, ajudando a evitar situações de sobreajuste e subajuste.

Por fim, no próximo capítulo, serão apresentados os gráficos com os melhores resultados de cada modelo. Esses gráficos proporcionarão uma comparação visual do desempenho de cada um e destacarão a importância da avaliação gráfica na avaliação do desempenho do modelo.

5 ANÁLISE DOS RESULTADOS

Este capítulo dedica-se à apresentação e discussão dos resultados durante a evolução deste trabalho, obtidos mediante a aplicação das técnicas de análise exploratória dos dados e da aplicação dos modelos de predição de carga elétrica sobre estes, utilizados neste estudo.

Inicialmente, será realizada uma análise aprofundada dos dados empregados, incluindo estudos descritivos e gráficos. Esta etapa permitirá a compreensão das características intrínsecas do conjunto de dados, suas distribuições e as inter-relações entre as variáveis.

Em seguida, serão apresentados os resultados decorrentes da análise dos modelos de predição. Esses resultados incluem métricas de validação, tempos de processamento e análises das curvas de perda e aprendizado dos modelos.

Será apresentada uma seção para discutir o impacto das características dos dados e dos recursos de *lookback* e janela deslizante no desempenho dos modelos. Esta análise contribuirá para a compreensão de como os diferentes componentes do modelo interagem e influenciam o resultado final.

Finalmente, será mostrada uma análise gráfica, que comparará os dados reais com os resultados previstos pelos modelos. Esta análise permitirá uma avaliação visual da capacidade dos modelos de prever a carga elétrica.

Cada seção será acompanhada por discussões dos resultados, com o intuito de ampliar o conhecimento e o senso crítico necessários para responder à Questão de Pesquisa. Este trabalho tem como objetivo principal o desenvolvimento de um algoritmo preditor de carga para os transformadores de potência da Eletrobras Eletronorte.

5.1 Resultados da Análises Exploratória dos Dados

Nesta seção, serão apresentados os resultados da Análise Exploratória dos Dados (AED), que constituiu um passo crucial na modelagem de predição. Como descrito no capítulo anterior, a AED permite conhecer melhor os dados, obter uma compreensão inicial dos mesmos e formular hipóteses sobre o comportamento do sistema que está sendo estudado.

Primeiramente, será efetuada uma análise descritiva do conjunto de dados, explorando suas características fundamentais, como médias, desvios padrão e valores máximos e mínimos. Esta análise proporcionará uma visão geral das tendências, padrões e irregularidades que podem ser encontradas nos dados.

A seguir, será realizada uma análise gráfica dos dados. Serão produzidos histogramas, boxplots e gráficos de dispersão que facilitarão a visualização das distribuições de dados e das correlações entre os atributos e eventuais *outliers* (os *outliers* da variável-alvo 'CARGA_PV' já foram eliminados num processo anterior, de pré-processamento da base de dados).

Posteriormente, serão discutidos os resultados dos testes de estacionariedade, autocorrelação e normalidade. Estes testes são fundamentais para verificar suposições específicas que muitos modelos estatísticos e de aprendizado de máquina fazem sobre os dados.

Cada uma dessas análises proporcionará uma compreensão mais profunda das características dos dados, que é vital para a construção de um modelo de predição preciso. Este processo de análise e entendimento dos dados constitui um ponto de partida essencial para a modelagem dos algoritmos de predição de carga elétrica.

5.2 Resultados das Análises Descritivas

A Análise Descritiva a seguir oferece uma visão inicial dos dados por meio do cálculo de medidas de tendência central, dispersão e forma, sendo possível identificar padrões básicos, como a média, a mediana, o desvio padrão, entre outros, que dão uma ideia inicial da estrutura dos dados. As Tabelas 12 e 13 ilustra os resultados da Análise Descritiva dos dados, por meio da utilização de biblioteca 'Pandas'.

Tabela 12 – Estatísticas descritivas do conjunto de dados

	CARGA_PV	tipo_dia	Precip	Umid. Relat. AR (%)	Temperatura do ar (°C)	Pressão Atm.
count	75991	75991	75991	75991	75991	75991
mean	174.014	0.678	0.355	82.535	26.084	975.503
std	32.368	0.467	2.305	15.548	3.331	1.948
min	66.361	0.000	0.000	27.000	14.500	967.900
25%	150.527	0.000	0.000	73.000	23.800	974.300
50%	172.280	1.000	0.000	88.000	25.100	975.600
75%	196.280	1.000	0.000	95.000	28.200	976.800
max	284.588	1.000	52.000	100.000	37.000	982.200

Fonte: do próprio autor.

Com base nos resultados apresentados, pode se observar que:

- CARGA_PV: Esta coluna representa a carga do transformador. Os dados indicam uma média de 174.01 MW, uma mediana de 172.28 MW, o que indica uma distribuição relativamente simétrica, sem influência de *outliers*, o que era esperado, vistos que estes foram eliminados no pré-processamento. A carga mínima observada foi de 66.36 MW e a máxima foi de 284.59 MW. Portanto, a carga do transformador parece variar

Tabela 13 – Medidas de Distribuição dos dados

	CARGA_PV	tipo_dia	Precip	Umid. Relat. AR (%)	Temperatura do ar (°C)	Pressão Atm.
média	174.014	0.678	0.355	82.535	26.084	975.503
mediana	172.280	1.000	0.000	88.000	25.100	975.600
moda	166.787	1.000	0.000	98.000	24.400	976.000
curtose	-0.098	-1.422	156.616	0.571	0.097	0.309
assimetria	0.215	-0.760	11.106	-1.120	0.671	-0.084

Fonte: do próprio autor.

bastante, mas a maioria dos valores está próxima da média. Sendo assim, uma vez que a capacidade máxima da subestação é de 400 MVA, observa-se que não houve episódio de sobrecarga durante o período dos dados.

- tipo_dia: Este atributo é uma variável binária (0 ou 1), representando dias úteis e não úteis. A média é de 0.68, indicando que a maioria dos dias são úteis (representados por 1). Além disso, o valor da moda também é 1, confirmando que a maior parte dos dias registrados são dias úteis.

- Precip: Este atributo representa a quantidade de precipitação. A média é de 0.36, o que indica que, em média, a precipitação é bastante baixa. No entanto, a moda e a mediana são ambas 0, sugerindo que a maioria dos dias não tem chuva. O valor máximo de precipitação é de 52, o que indica a ocorrência de eventos de precipitação extremamente elevados, mas raros. O valor da curtose (156.62) é extremamente alto, indicando uma distribuição de valores altamente concentrada em torno da mediana, com caudas muito pesadas. Isso significa que houve alguns dias com chuvas muito pesadas que são bastante incomuns em comparação com o padrão geral de chuvas na região. O valor da assimetria (11.11) é bastante alto e positivo, indicando que a cauda direita da distribuição é muito mais longa ou há valores muito mais extremos no lado direito da distribuição. Isso está em linha com a observação da curtose e sugere que houve alguns dias com chuvas muito pesadas. Em termos práticos, interpreta-se que, embora haja alguns eventos de chuva pesada, a maior parte do tempo é seca.

- Umid. Relat. AR (%): Esta coluna representa a umidade relativa do ar. A média de 82.53% e a mediana de 88% indicam que a umidade é geralmente alta. Além disso, a moda é 98%, o que sugere que muitos dias experimentam alta umidade, o que é uma característica marcante do clima da região amazônica.

- Temperatura do ar (C): A média da temperatura do ar é de 26.08°C, com uma mediana de 25.1°C e uma moda de 24.4°C. Isso sugere que a temperatura é geralmente estável e quente.

- Pressão Atm.: A pressão atmosférica tem uma média de 975.5 hPa, uma mediana de 975.6 hPa e uma moda de 976 hPa, indicando que a pressão atmosférica também é bastante estável.

Além disso, com base nos valores de curtose e assimetria, podemos inferir que:

- A curtose dos dados de carga (CARGA_PV), tipo de dia (tipo_dia) e precipitação (Precip) é negativa, o que indica que a distribuição é mais plana do que a distribuição normal. A curtose para umidade, temperatura e pressão é positiva, o que indica uma distribuição mais afunilada do que a distribuição normal.
- A assimetria dos dados de carga (CARGA_PV) e temperatura (Temperatura do ar) é positiva, o que indica que a cauda à direita da distribuição é mais pesada ou mais longa. Para os demais atributos, a assimetria é negativa, o que indica que a cauda à esquerda da distribuição é mais pesada ou mais longa.

Finalmente, a Tabela 14 traz a correlação entre as variáveis do conjunto de dados utilizado.

Tabela 14 – Correlação entre os atributos

	CARGA_PV	tipo_dia	Precip	Umid. Relat. AR (%)	Temperatura do ar (C)	Pressão Atm.
CARGA_PV	1.000	0.187	-0.020	-0.131	0.276	-0.454
tipo_dia	0.187	1.000	0.009	0.004	0.011	-0.059
Precip	-0.020	0.009	1.000	0.131	-0.123	0.026
Umid. Relat. AR (%)	-0.131	0.004	0.131	1.000	-0.832	0.037
Temp. do ar (C)	0.276	0.011	-0.123	-0.832	1.000	-0.296
Pressão Atm.	-0.454	-0.059	0.026	0.037	-0.296	1.000

Fonte: do próprio autor.

De acordo com a tabela anterior, algumas observações podem ser realizadas:

i. CARGA_PV e tipo_dia: Há uma correlação positiva de 0.186641, o que indica uma relação linear fraca e direta. Isso sugere que o tipo de dia tem algum efeito sobre a carga de energia, mas não é um determinante forte.

ii. CARGA_PV e Temperatura do ar (C): Há uma correlação positiva de 0.276206, o que sugere que há uma relação linear fraca entre a carga e a temperatura do ar.

iii. CARGA_PV e Pressão Atm.: Há uma correlação negativa de -0.453826, que é uma correlação linear moderada. Isso significa que quando a pressão atmosférica aumenta, a carga tende a diminuir.

iv. Umid. Relat. AR (%) e Temperatura do ar (C): Há uma correlação negativa forte de -0.831554, o que indica que quando a umidade relativa do ar aumenta, a temperatura do ar tende a diminuir, e vice-versa.

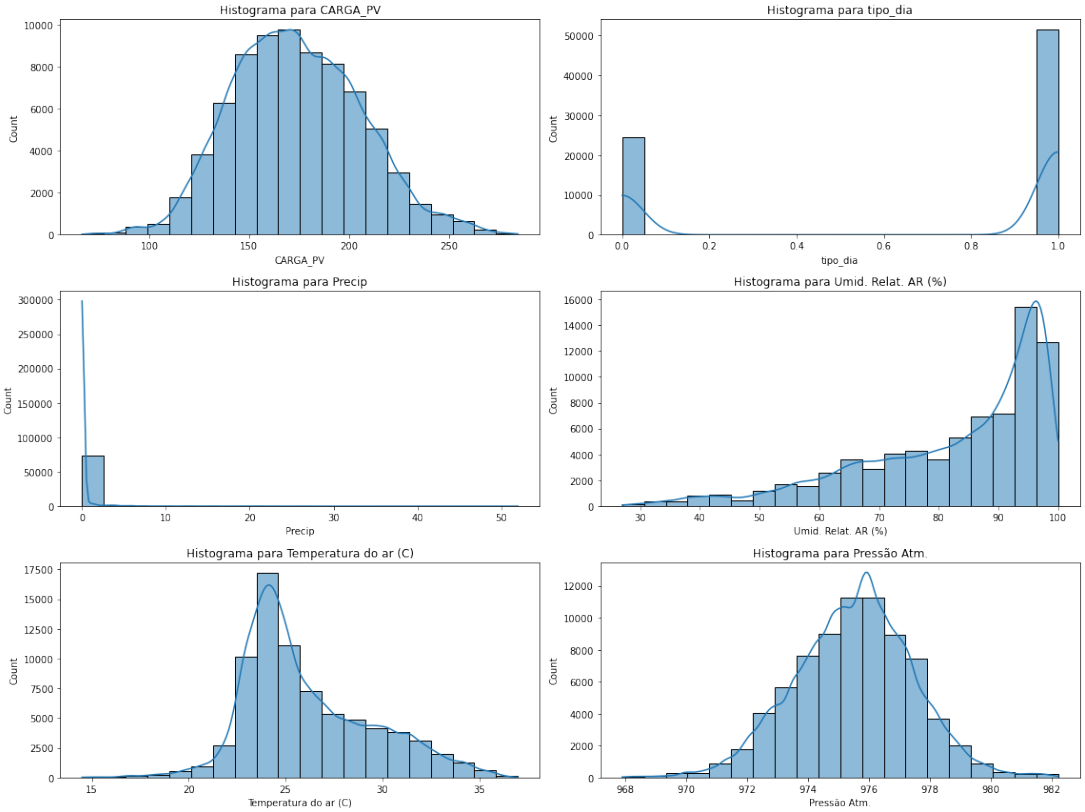
v. Todos os outros pares de variáveis possuem correlações muito fracas (próximas a zero), o que indica que não há uma relação linear forte entre eles.

As análises descritivas apresentaram características essenciais e as relações entre os atributos do conjunto de dados. Complementarmente, a próxima seção deste capítulo irá permitir a visualização destas características e relações, de forma a reforçar a compreensão dos dados trabalhados.

5.3 Resultado das Análises Gráficas

A Figura 23 apresenta os histogramas e a Figura 24 apresenta os *bloxplots* dos atributos do conjunto de dados estudado. As Figuras 25 e 26 apresentam a correlação entre eles. E, finalmente, A Figura 27 apresenta a variação temporal dos atributos ao longo do tempo da janela de dados amostrados e consolidados.

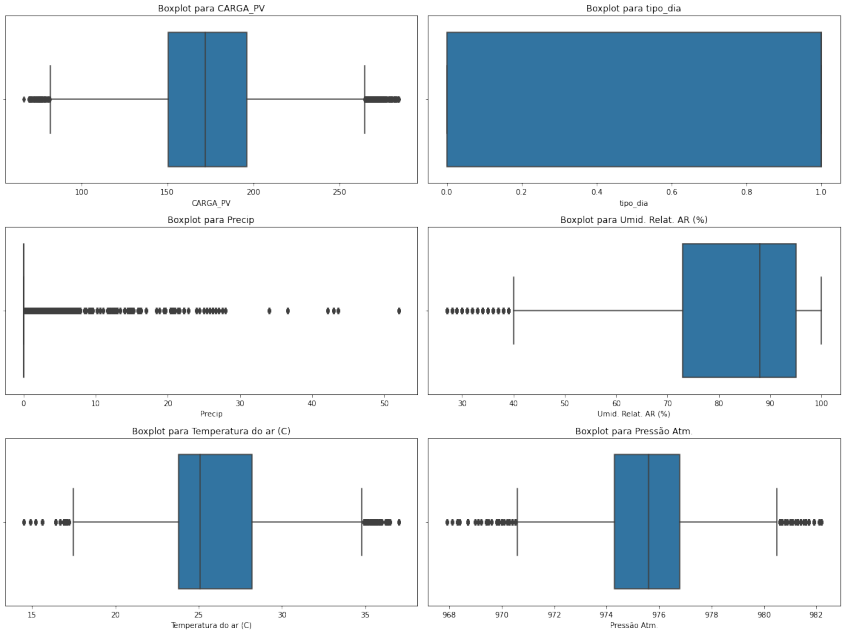
Figura 23 – Histograma dos atributos do conjunto de dados



Fonte: do próprio autor.

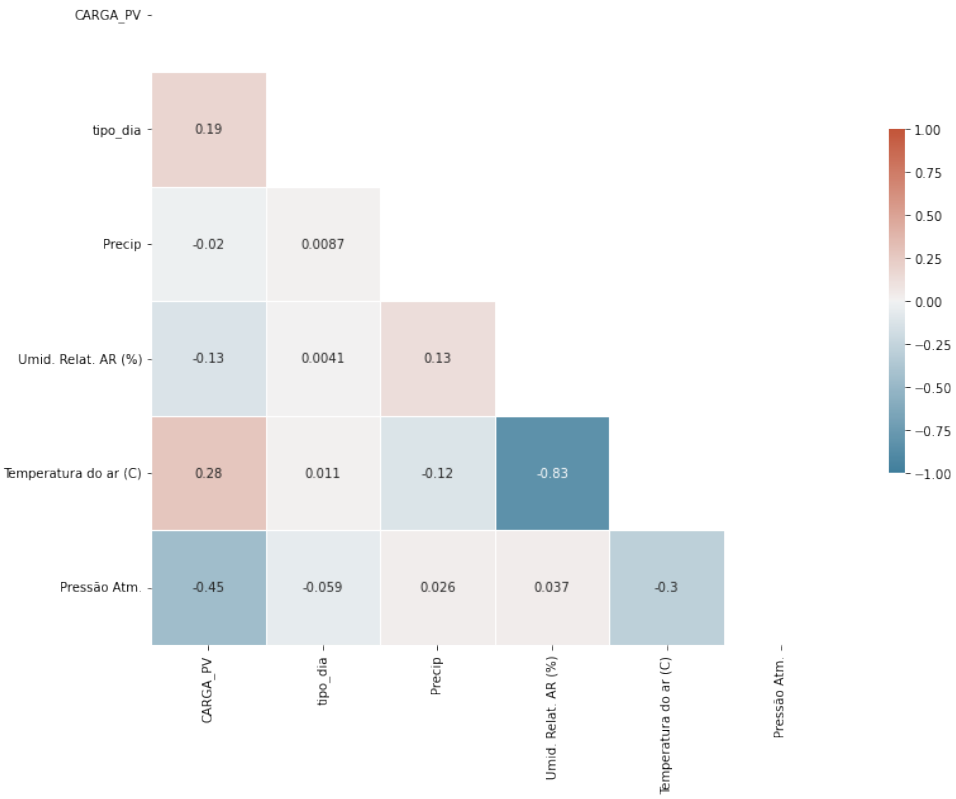
Conforme observado nas figuras desta seção, as verificações realizadas no tópico anterior, sobretudo aquelas que dizem respeito à curtose e à assimetria dos dados, per-

Figura 24 – Boxplot dos atributos do conjunto de dados



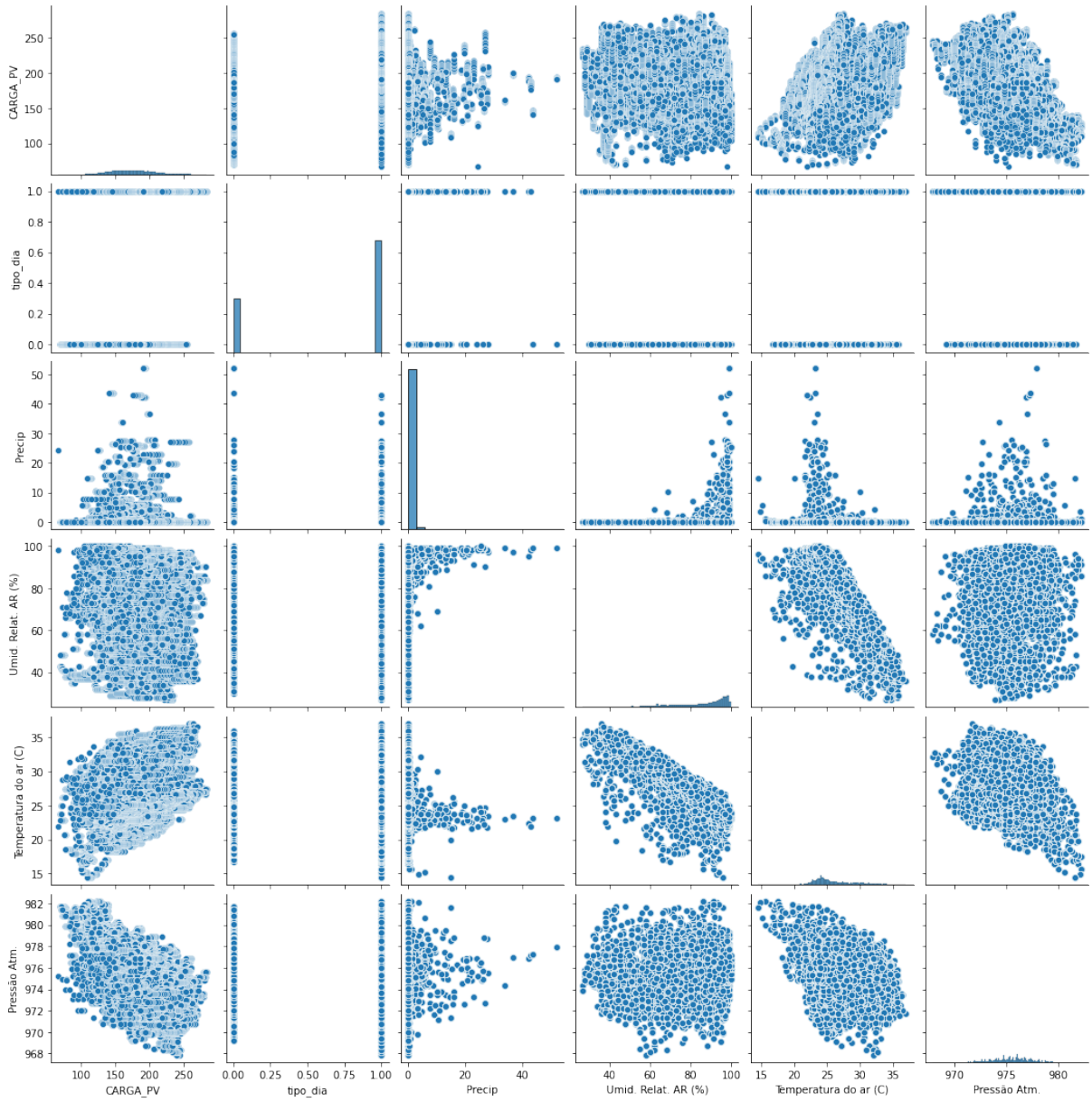
Fonte: do próprio autor.

Figura 25 – Correlação entre os atributos do conjunto de dados



Fonte: do próprio autor.

Figura 26 – Gráfico de dispersão das variáveis do conjunto de dados

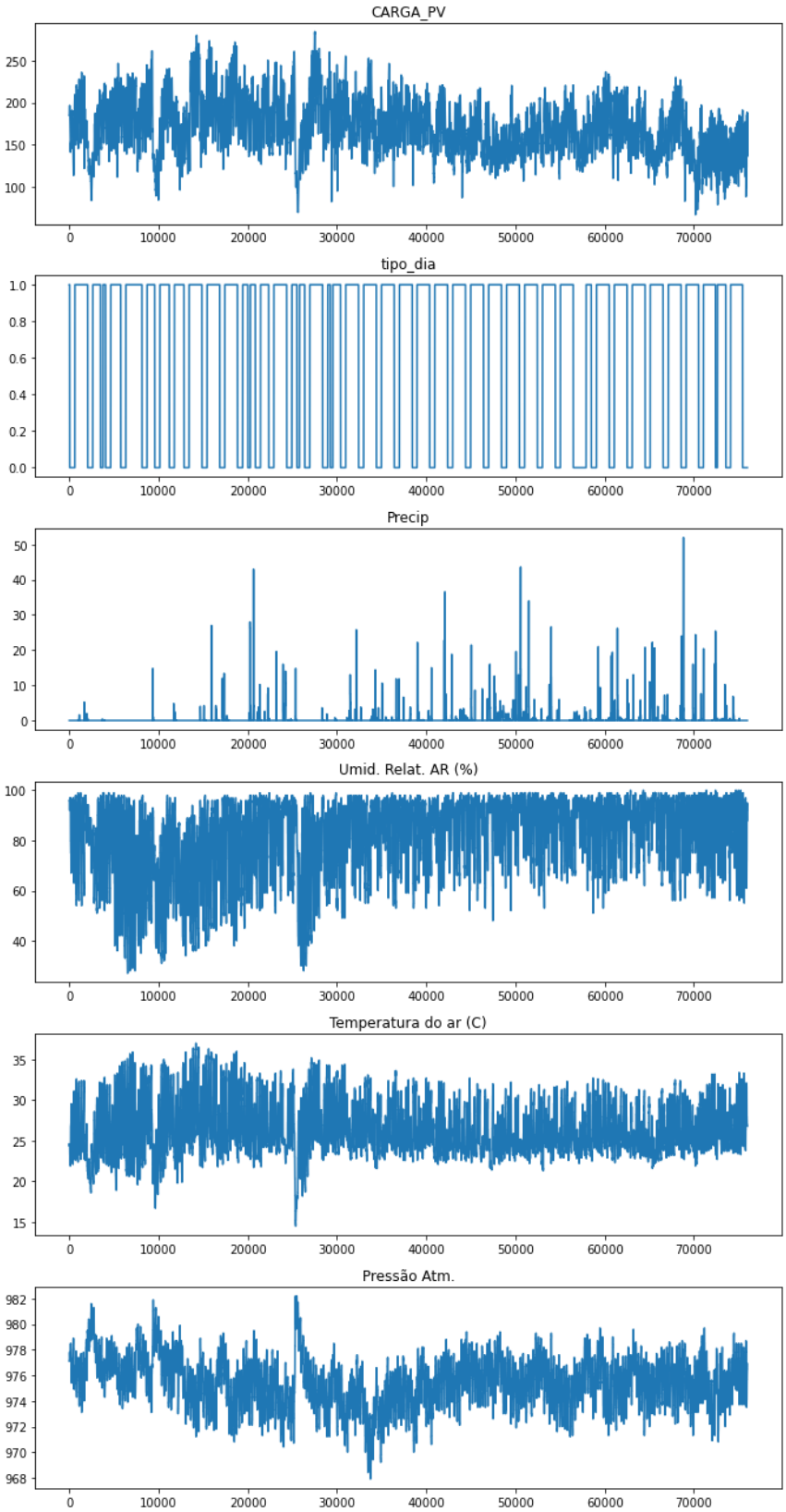


Fonte: do próprio autor.

manecem válidas. Com relação ao gráfico de dispersão, é facilmente possível verificar a correlação positiva entre a temperatura e a carga, bem como a correlação negativa entre temperatura e umidade, pressão atmosférica e temperatura e, finalmente, carga e pressão atmosférica - também validando as observações numéricas da análise descritiva anterior.

No próximo item deste capítulo serão apresentados e discutidos os resultados dos testes de estacionariedade, de autocorreção e de normalidade, muito importantes para certificar se os dados podem se ajustar aos modelos de predição selecionados neste trabalho.

Figura 27 – Variação temporal dos atributos do conjunto de dados



Fonte: do próprio autor.

5.4 Resultados do Teste de Estacionariedade

Nesta seção, por meio da Tabela 15, será apresentado o resultado do teste de estacionariedade, através da rotina *adfuller* da biblioteca python *statsmodels* sobre a variável 'CARGA_PV' conjunto de dados. Conforme abordado no capítulo anterior, certificar-se da estacionariedade da série temporal é um passo importante para garantir a modelagem adequada do preditor.

Tabela 15 – Resultado do Teste Dickey-Fuller Aumentado para CARGA_PV

Item	**Valor**
ADF *test statistic*	-19.336435
p-value	0.0
# lags used	64
# observations	75926
critical value (1%)	-3.430436
critical value (5%)	-2.861578
critical value (10%)	-2.566790

Fonte: do próprio autor.

Com base nestes resultados, visto que o valor do teste ADF é significativamente menor que dos valores críticos de 1%, 5% e 10% e que o valor-p é igual a zero, rejeita-se a hipótese nula de que a série seja não-estacionária. Portanto, o resultado indica que 'CARGA_PV' seja uma série temporal estacionária.

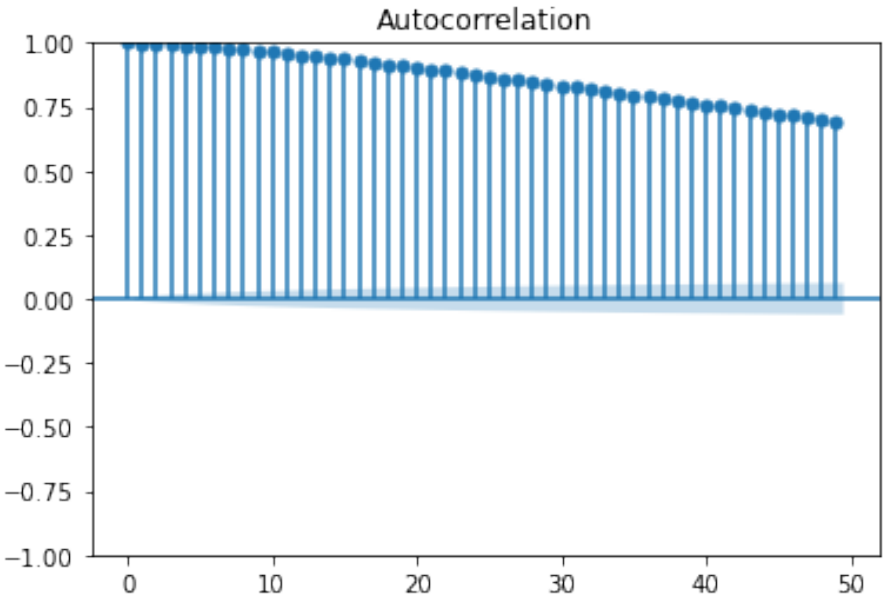
5.5 Resultados dos Testes de Autocorrelação

Seguindo com a análise exploratória dos dados, nesta seção será apresentados os testes de autocorrelação sobre a variável 'CARGA_PV' do conjunto de dados utilizado. Conforme destacado no capítulo anterior, foram realizados os testes de Função de Autocorrelação (ACF), Função de Autocorrelação Parcial (PACF), Gráfico de Latência, teste de Durbin-Watson e teste de Ljung-Box. A importância de verificar a autocorrelação em uma série temporal reside na sua capacidade de avaliar a interdependência entre as amostras, orientar a decisão sobre o uso de amostras anteriores na previsão das futuras e auxiliar na escolha do modelo preditor mais adequado.

A Figura 28 é a saída do teste ACF, a Figura 29 é a saída do PACF e a Figura 30, a saída do Gráfico de Latência. As Tabelas 16 e 17 trazem os resultados dos testes de Durbin-Watson e de Ljung-Box, respectivamente.

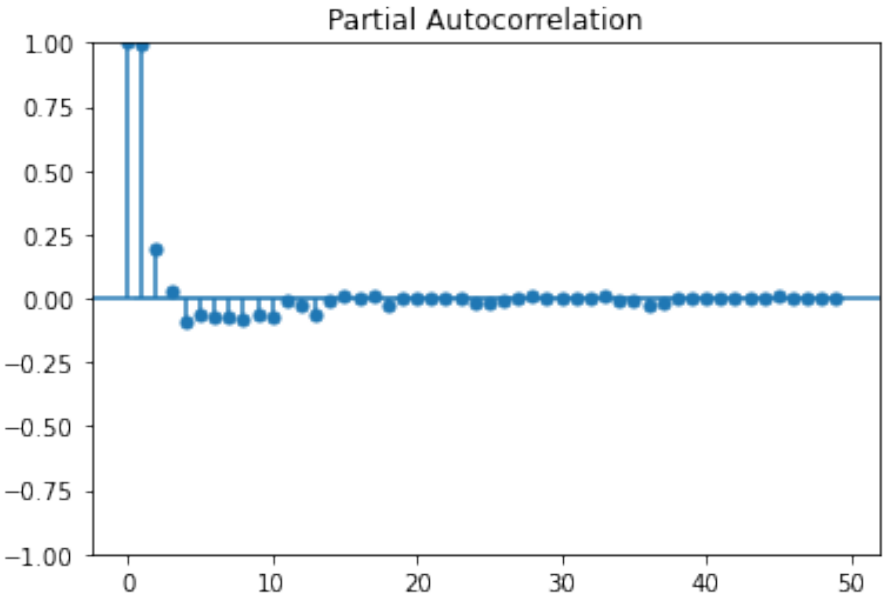
Do teste ACF, da Figura 28, verifica-se um decaimento lento. Este comportamento é característico de séries temporais que têm uma forte dependência de valores anteriores, onde a correlação entre um ponto e seus pontos anteriores não diminui rapidamente. Do teste de PACF, da Figura 29, observa-se um decaimento logarítmico, indicando que a

Figura 28 – Resultado do Teste ACF



Fonte: do próprio autor.

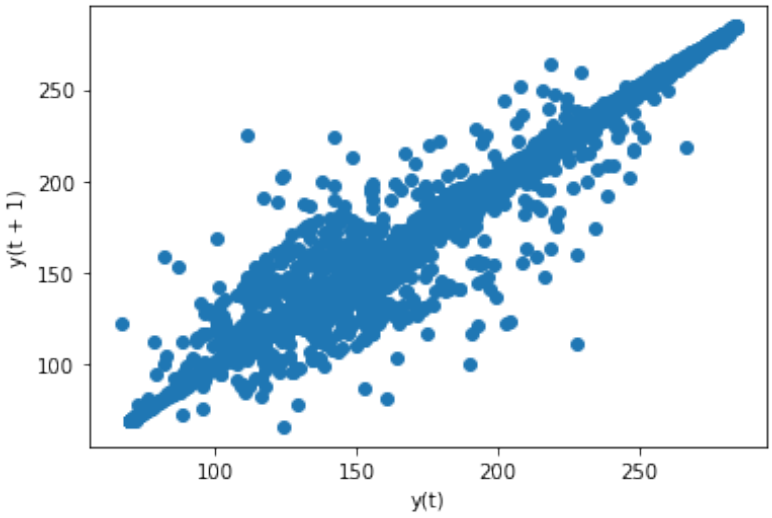
Figura 29 – Resultado do Teste PACF



Fonte: do próprio autor.

autocorrelação entre a observação atual e suas observações anteriores diminui em uma taxa logarítmica. Isso pode sugerir que a autocorrelação tem um efeito de longo alcance na série, mas sua força diminui com atrasos maiores. Já do gráfico de latência (Figura 30), é possível identificar que os pontos formam uma diagonal crescente, ainda que com alguns pontos periféricos. Este comportamento indica uma forte correlação positiva entre uma observação e a sua defasagem, embora que os pontos fora da diagonal sugira uma leve

Figura 30 – Gráfico de Latência da variável CARGA_PV



Fonte: do próprio autor.

Tabela 16 – Resultado do teste de Durbin-Watson

Teste	**Resultado**
Durbin-Watson	0.0002500223924117734

Fonte: do próprio autor.

Tabela 17 – Resultado do teste de Ljung-Box para 10 lags

Medida	**Resultado**
lb_stat 10	733006.462488
lb_pvalue 10	0.0

Fonte: do próprio autor.

não-linearidade.

Da Tabela 16, o resultado do teste de Durbin-Watson apresentou valor próximo de zero, sugerindo uma autocorrelação positiva muito forte. O teste de Ljung-Box, na Tabela 17, apresentou um lb_stat foi muito alto e o valor do lb_pvalue foi 0. Isso indica que se rejeita a hipótese nula de que os dados são independentes (não autocorrelacionados). Ou seja, confirmamos que há autocorrelação nos dados.

Portanto, com base nos resultados dos testes gráficos (ACF, PACF e de Latência) e nos testes de Durbin-Watson e Ljung-Box, podemos afirmar com segurança que a variável 'CARGA_PV' do conjunto de dados apresenta forte autocorrelação. Dessa forma, estão habilitados para a utilização modelos autoregressivos, bem como as LSTM e os GRUs, que já são projetados para lidar com a sequência de dados com algum grau de lembrança. Adicionalmente, atesta-se a utilização de técnicas que utilizam de informações passadas para prever valores futuros, como são as janelas deslizantes e o *look-back*.

5.6 Resultados dos Testes de Normalidade

Este tópico visa apresentar os resultados dos testes de normalidade, conforme indicado no capítulo anterior: Os testes de Shapiro-Wilk e Anderson-Darling. Reitera-se a importância de testar a normalidade dos dados para garantir a validade e a precisão das inferências estatísticas realizadas a partir de modelos que assumem esta propriedade. Novamente, a normalidade é uma suposição central para diversos modelos de previsão de séries temporais, e se os dados não seguem uma distribuição normal, algumas alternativas devem ser consideradas, como a aplicação de transformações de dados ou a escolha de modelos que não fazem tal suposição.

Assim, juntamente com os demais testes (de estacionariedade e de autocorrelação), temos as melhores condições para a escolha adequada dos modelos preditores, isto é, um modelo preditor que melhor se ajuste às características dos dados trabalhados. A Tabela 18 apresenta os resultados dos testes de Shapiro-Wilk e Anderson-Darling, respectivamente.

Tabela 18 – Resultados dos testes de normalidade

Teste	***Statistic***	***p-value***
Shapiro-Wilk	0.996	0.000
Anderson-Darling	84.237	-

Fonte: do próprio autor.

De acordo com estes resultados, o valor estatístico do Teste de Shapiro-Wilk é bastante próximo de 1 e sugere fortemente que os dados são normalmente distribuídos. No entanto, o valor-p é 0.000, que é menor que o nível de significância comum de 0.05. Isso implica que rejeitamos a hipótese nula de que a população é normalmente distribuída. Portanto, com base no Teste de Shapiro-Wilk, concluímos que os dados não seguem uma distribuição normal. O Teste de Anderson-Darling, por sua vez, apresentou um valor estatístico de 84.237, que é bastante alto, sugerindo que os dados não seguem uma distribuição normal. Assim, o Teste de Anderson-Darling também indica a rejeição da hipótese de que a população tem uma distribuição normal. Em suma, ambos os testes indicam que os dados testados (a variável 'CARGA_PV do conjunto de dados) não são normalmente distribuídos.

Este resultado implica na escolha dos modelos preditores, com base na premissa de que alguns dos modelos estatísticos, como o ARMA e o ARIMA, assumem a normalidade de dados. Isto é, se a suposição de normalidade não for atendida, as inferências feitas a partir desses modelos podem ser imprecisas.

Portanto, a escolha dos modelos preditores adotados no trabalho se fundamentam na premissa de que a normalidade dos dados não seja uma dependência inerente - que sejam flexíveis em relação à distribuição dos dados, como são as Redes Neurais, as SVMs

e os modelos baseados em árvores de decisão.

Sendo assim, no próximo item serão apresentados os resultados da aplicação destes modelos sobre o conjunto de dados, no intuito de avaliar qual modelo obteve as melhores métricas de avaliação sobre o resultado da previsão.

5.7 Resultados da Aplicação dos Modelos sobre o conjunto de dados

Nesta seção serão apresentados os resultados das métricas de avaliação dos modelos adotados neste trabalho, quais foram: os modelos de Regressão Linear e Polinomial, a SVM, os modelos baseados em árvore de decisão Floresta Aleatória e *XGBoost* e as Redes Neurais ANN, CNN, LSTM e GRU. Mais adiante, serão apresentadas as curvas de aprendizado dos modelos baseados em Regressão, SVM e Árvore de Decisão, bem como as curvas de perda para aqueles modelos baseados em Redes Neurais. Finalmente, serão apresentados os gráficos de previsão de cada modelo, comparando-os com os respectivos dados reais.

Conforme descrito detalhadamente no capítulo anterior, todos os modelos foram submetidos à mesma base de treinamento e teste, bem como antes da aplicação de cada modelo, o conjunto de dados passou pelo mesmo processo de normalização de dados, a fim de garantir a consistência da avaliação de comparação entre os modelos.

5.7.1 Avaliação das Métricas de Desempenho dos Modelos

A seguir, os resultados de R^2 e MSE para cada modelo aplicado sobre o conjunto de dados, bem como o tempo gasto para o processamento do treinamento e previsão, todos compilados na Tabela 19.

Tabela 19 – Desempenho dos Modelos de Previsão

Modelo	**Métricas de desempenho**		**Treinamento**		**Previsão**	
	R^2	**MSE**	**CPU times**	**Wall times**	**CPU times**	**Wall times**
Regressão Linear	0.980	12.21	62.5 ms	12 ms	15.6 ms	3.99 ms
Regressão Polinomial	0.964	25.51	2.59 s	1.38 s	46.9 ms	47.9 ms
SVM (*kernel*: rbf)	0.981	13.46	7min 11s	7min 14s	1min 31s	1min 32s
SVM (*kernel*: poly)	0.976	16.77	3min 58s	3min 58s	19.4 s	19.4 s
Random Forest	0.984	11.23	18.4 s	18.7 s	375 ms	383 ms
XGBoost	0.982	12.44	38 s	5.9 s	93.8 ms	22.9 ms
ANN	0.981	13.49	4min 2s	2min 35s	938 ms	1.03 s
CNN	0.982	12.83	3min 1s	1min 41s	1.23s	1.01s
LSTM	0.982	12.69	1h 15min 25s	20min 41s	9.12 s	4.49 s
GRU	0.978	16.41	54min 46s	16min 41s	4.11s	2.21s

Fonte: do próprio autor.

Com base nos resultados acima, algumas observações podem ser destacadas, quais sejam:

i. Todos os modelos apresentaram bom desempenho em termos de R^2, com valores acima de 0.96, o que indica um bom ajuste dos modelos aos dados.

ii. O modelo de Regressão Linear teve o tempo de treinamento e previsão mais curto, tornando-o o modelo mais eficiente em termos de velocidade de processamento.

iii. A Regressão Polinomial, apesar de ter um desempenho inferior comparado aos outros modelos em termos de R2 e MSE, ainda teve uma performance aceitável. Entretanto, este modelo tem ao seu favor os tempos de treinamento e previsão que, ainda que superiores aos da Regressão Linear, são muito satisfatórios, o que o torna uma opção promissora.

iv. O modelo SVM com *kernel* polinomial apresentou um desempenho ligeiramente inferior em comparação com o modelo SVM com *kernel* RBF, indicando que a escolha do *kernel* pode afetar significativamente o desempenho do SVM.

v. O modelo SVM, especialmente com o *kernel* RBF, apresentou um bom desempenho em termos de métricas de erro (R2 e MSE), porém com tempos de processamentos muito mais longos em comparação a maioria dos modelos aplicados, o que pesa contra seu uso prático se comparado com o desempenho de outros modelos.

vi. Em termos do Erro Quadrático Médio (MSE), o modelo de Floresta Aleatória (*Random Forest*) obteve o menor valor, indicando que ele tem a menor taxa de erro na previsão.

vii. O modelo XGBoost, apesar de não ter o menor MSE, apresentou um tempo de treinamento relativamente rápido em comparação com outros modelos com desempenho semelhante.

viii. Entre os modelos baseados em redes neurais, a ANN (*Artificial Neural Network*) e a CNN (*Convolutional Neural Network*) tiveram um desempenho semelhante em termos de R2 e MSE. Porém, o tempo de treinamento da ANN foi superior, sugerindo que a CNN pode ser uma opção mais eficiente para este conjunto de dados.

ix. A ANN apresentou um bom desempenho de previsão, ficando na média dos modelos em termos de R2 e MSE. O tempo de previsão, em termos práticos, também é baixo. Isso sugere que, para este conjunto de dados, as redes neurais artificiais são uma opção de preditor. Entretanto, o tempo gasto com o treinamento do modelo o torna uma das últimas opções.

x. A Rede Neural Convolucional (CNN) teve um bom desempenho em termos das métricas de erro utilizadas (R2 e MSE), com valores próximos aos melhores modelos. No entanto, o tempo de treinamento da CNN foi maior em comparação com alguns outros modelos, com desempenho semelhante. Isso é típico das redes neurais, que muitas vezes exigem mais tempo de processamento devido à sua complexidade e ao número de parâmetros a serem otimizados.

xi. O modelo LSTM teve um tempo de treinamento muito longo, mas ainda assim conseguiu um bom desempenho. No entanto, o longo tempo de treinamento pode torná-lo menos preferível para conjuntos de dados muito grandes. Uma arquitetura menos robusta pode torna-lo mais rápido e assim colocá-lo numa opção mais viável, ainda que seja esperada alguma diminuição nas métricas de desempenho. Comentários adicionais serão realizados após a apresentação da Curva de Perda deste modelo.

xii. A GRU gastou um pouco menos de tempo de processamento para o treinamento do modelo em relação ao LSTM, com desempenho em termos de métrica também ligeiramente inferior. Por outro lado, o tempo utilizado para a previsão foi bem superior ao outro modelo Recorrente. Se comparado com os demais, o tempo de treinamento do modelo o coloca entre as última opções. Assim como a LSTM, merece uma análise *a posteriori* com uma arquitetura menos robusta e com tempos de treinamentos inferiores se o desempenho do modelo permanece satisfatório.

5.7.2 Avaliação das Curvas de Aprendizado e Perda dos modelos

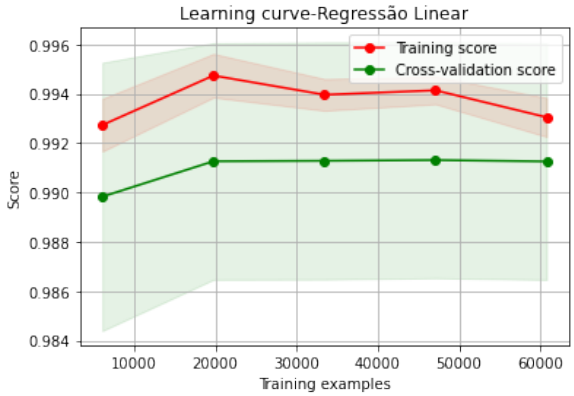
A fim de complementar as informações sobre o processo de treinamento e teste dos modelos, bem como do aprendizado-ajustes dos mesmos, nas Figuras 31 e 32 são apresentadas as curvas de aprendizado e perda, segundo a metodologia empregada em cada modelo. Destas figuras, é possível realizar algumas considerações para cada modelo:

i. Regressão Linear: Esse modelo teve um desempenho muito bom em termos de *score*, atingindo cerca de 0.99 em ambas as fases de treinamento e validação. O desvio padrão baixo indica que o modelo foi consistente em seus resultados. Entretanto, observa-se que não houve melhoria significativa na pontuação de validação com o aumento do tamanho dos dados de treinamento. Isso sugere que o modelo pode ter alcançado sua capacidade máxima de aprendizado, não sendo necessário um conjunto de dados superior a 20k amostras para obter uma previsão satisfatória.

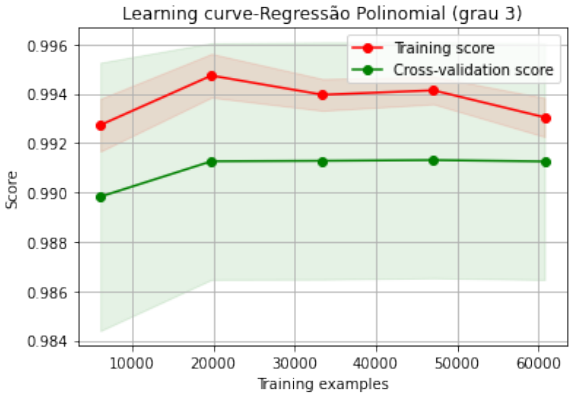
ii. Regressão Polinomial: Semelhante ao modelo de Regressão Linear, a Regressão Polinomial também apresentou resultados muito bons com *scores* em torno de 0.99. Entretanto, assim como a Regressão Linear, o aumento do tamanho dos dados de treinamento não resultou em melhorias significativas na pontuação de validação.

iii. SVM, *kernel* rbf: Este modelo apresentou um dos maiores *score* de treinamento entre todos, chegando a cerca de 0.996, mas apresentou o menor *score* de validação quando o tamanho dos dados de treinamento era baixo (cerca de 0.92). Isso pode indicar um *overfitting* inicial, pois o modelo estava aprendendo muito bem os dados de treinamento, mas não estava generalizando bem para os dados de validação. À medida que o tamanho dos dados de treinamento aumentou, o *score* de validação

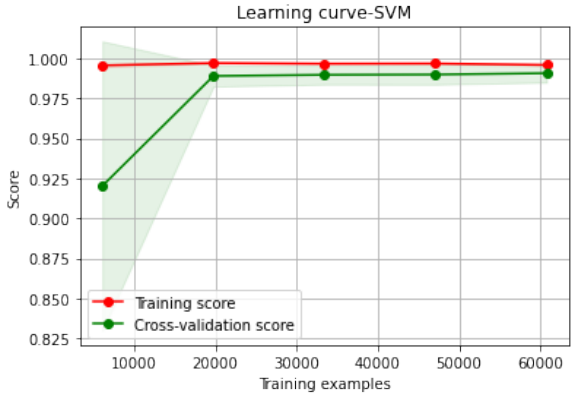
Figura 31 – Curvas de Aprendizado



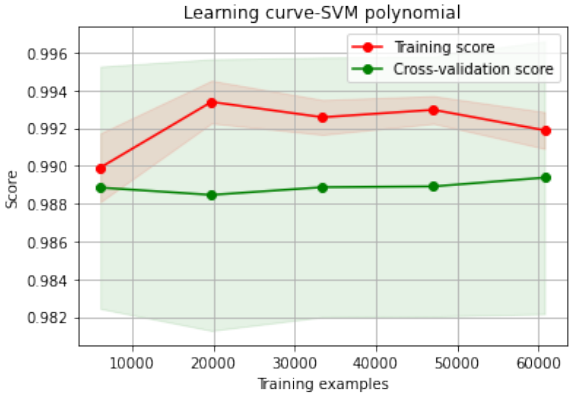
(a) Regressão Linear



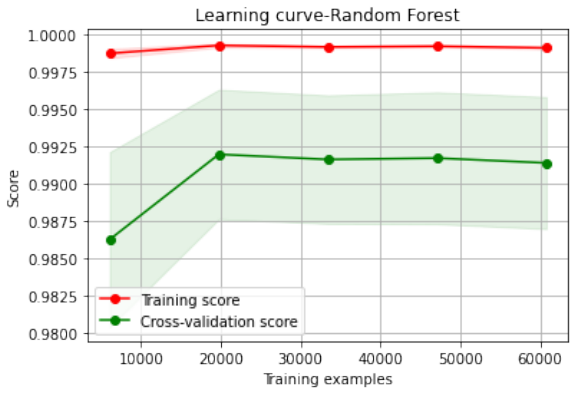
(b) Regressão Polinomial



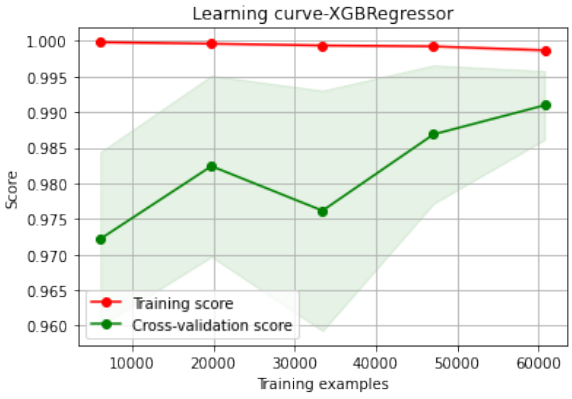
(c) SVM, *kernel* RBF



(d) SVM, *kernel* polynomial



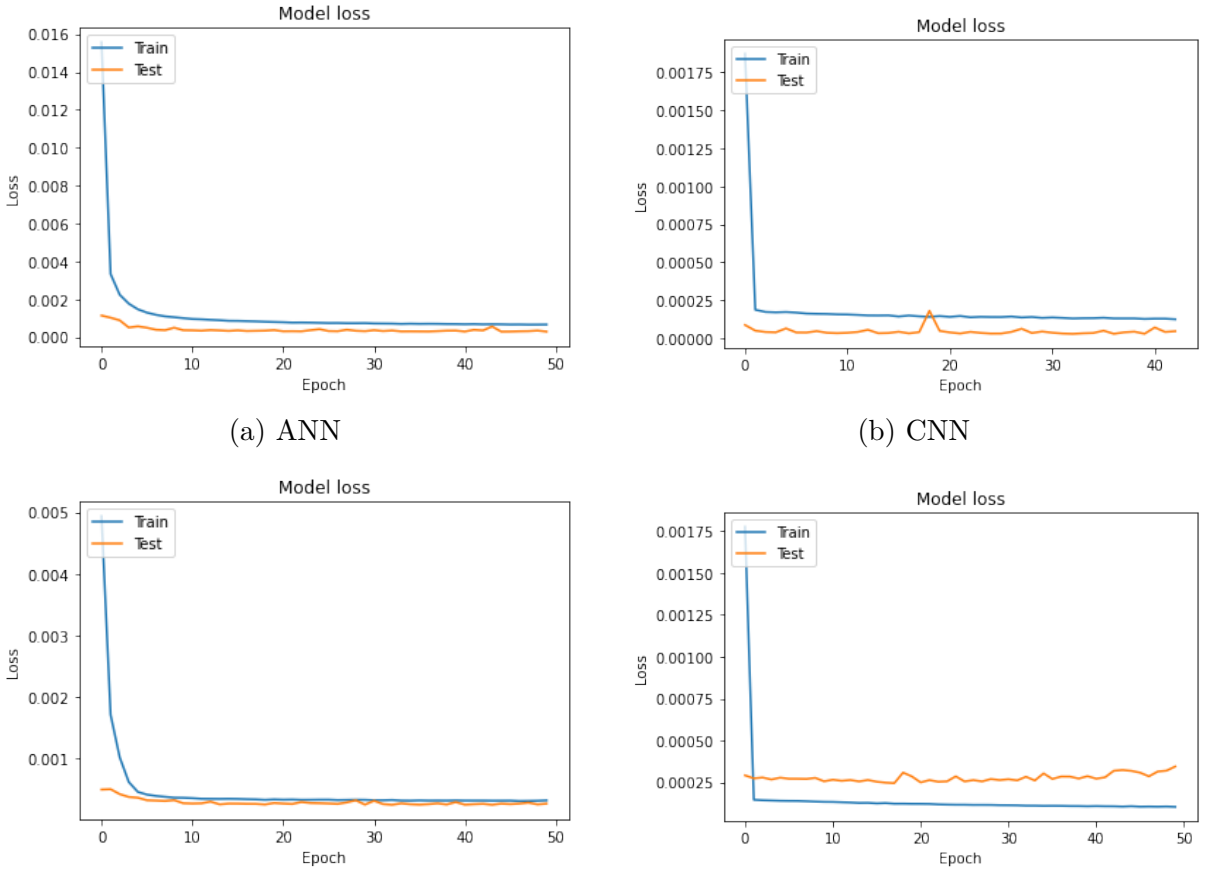
(e) *Random Forest*



(f) XGBoost

Fonte: do próprio autor.

Figura 32 – Curva de perda dos modelos baseados em Rede Neural



(a) ANN (b) CNN

(c) LSTM (d) GRU

Fonte: do próprio autor.

melhorou significativamente, indicando que o modelo se beneficiou do aumento da quantidade de dados de treinamento.

iv. SVM, *kernel* poly: Este modelo apresentou um desempenho inferior em relação ao SVM com *kernel* rbf, tanto em termos de *scores* de treinamento quanto de validação. Inicialmente, as curvas de treinamento e validação começaram muito próximas uma da outra, mas com um *score* menor ou igual a 0.99, afastando-se à medida da progressão do conjunto de dados. A partir das 20k amostras, as curvas de treinamento e validação começam a convergir novamente, indicando que o modelo está generalizando bem.

v. *Random Forest*: Este modelo apresentou o maiores *score* de treinamento, indicando um ajuste quase perfeito aos dados de treinamento. No entanto, seu *score* de validação foi ligeiramente inferior em comparação com sua curva de treinamento, embora ainda alto. Essa diferença, ainda que visualmente significativa, não indica *overfitting*, visto que o *score* da curva de cross-validation é alto ao longo de toda a sua progressão, indicando boa generalização por todo o conjunto de dados. Assim como nas curvas

dos modelos de Regressão e no SVM *kernel* RBF, um conjunto de 20k amostras parece ser ideal para realizar a predição dos dados.

vi. XGBoost: Este modelo também apresentou um alto *score* de treinamento, mas seu *score* de validação foi o menor entre todos os modelos quando o tamanho dos dados de treinamento era baixo. No entanto, o XGBoost mostrou melhorias significativas na pontuação de validação com o aumento dos dados de treinamento, sugerindo que se beneficiou com mais dados.

Das curvas de perda da Figura 32, dos modelos baseados em Redes Neurais, os seguintes comentários:

vii. ANN: A curva de perda de treinamento cai exponencialmente, o que indica que o modelo está aprendendo com os dados de treinamento muito rapidamente. No entanto, a curva de perda de teste já inicia num valor baixo e permanece assim ao longo das épocas com pequenas variações. Rapidamente as curvas de treinamento e teste se aproximam, tangenciando-se uma a outra. Este comportamento denota que o modelo está conseguindo se ajustar bem aos dados de treinamento e minimizar a diferença entre as previsões do modelo e os valores reais, generalizando bem para novos dados.

viii. CNN: A curva de perda de treinamento cai abruptamente, ainda mais rápido que a ANN, indicando que o modelo generaliza muito bem para o conjunto de dados utilizado. Aparentemente, a curva de treinamento descolada da curva de teste pode dar uma falta impressão de *underfitting*, entretanto, ao perceber o valor absoluto para o qual as duas curvas convergem, percebe-se que a generalização dos dados é muito boa, afastando a sensação de subajuste do modelo.

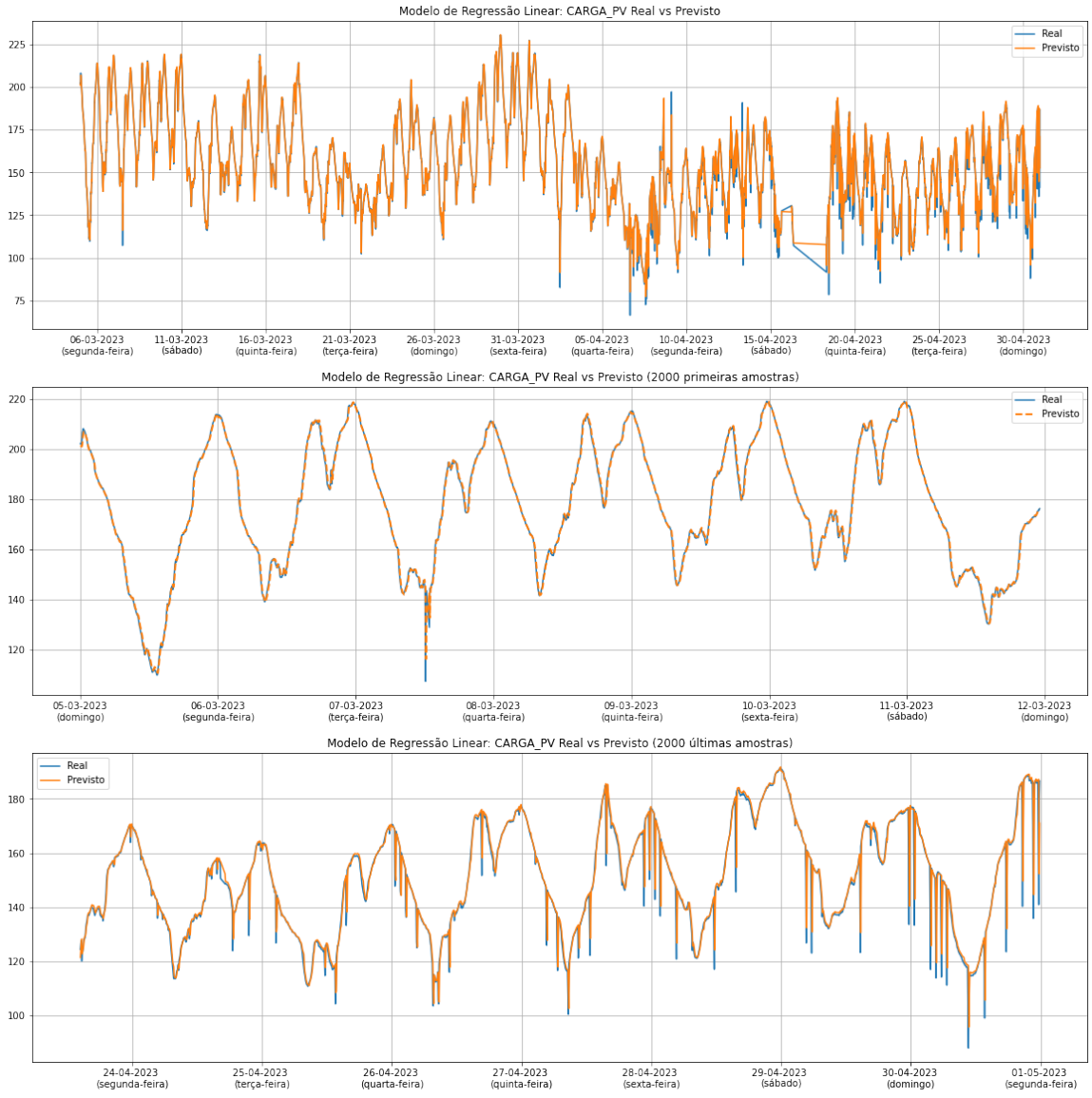
ix. LSTM: Graficamente, a curva da LSTM se assemelha com a do ANN, com um acoplamento entre as curvas mais rápido e mais justo, indicando um ótimo ajuste dos dados, sem *over* ou *underfitting*, com ótima generalização dos dados. Ambas as curvas convergem rapidamente num valor de perda muito baixo, indicando que não é necessário tantas épocas para o treinamento deste modelo e assim, reduzir o excessivo tempo de processamento observado na Tabela 19.

x. GRU: A curva de perda de treinamento e teste apresentam ótima convergência, ambas com valores de perda extremamente baixos. Embora novamente exista uma discrepância visual, agora com a curva de teste sendo ligeiramente superior à de treinamento na maior parte das épocas, a diferença é minimizada dada a baixa escala dos valores. Ou seja, o modelo GRU demonstra um alto desempenho tanto no treinamento quanto no teste, sugerindo um aprendizado e generalização eficazes dos dados.

5.7.3 Avaliação Gráfica dos resultados de previsão dos modelos

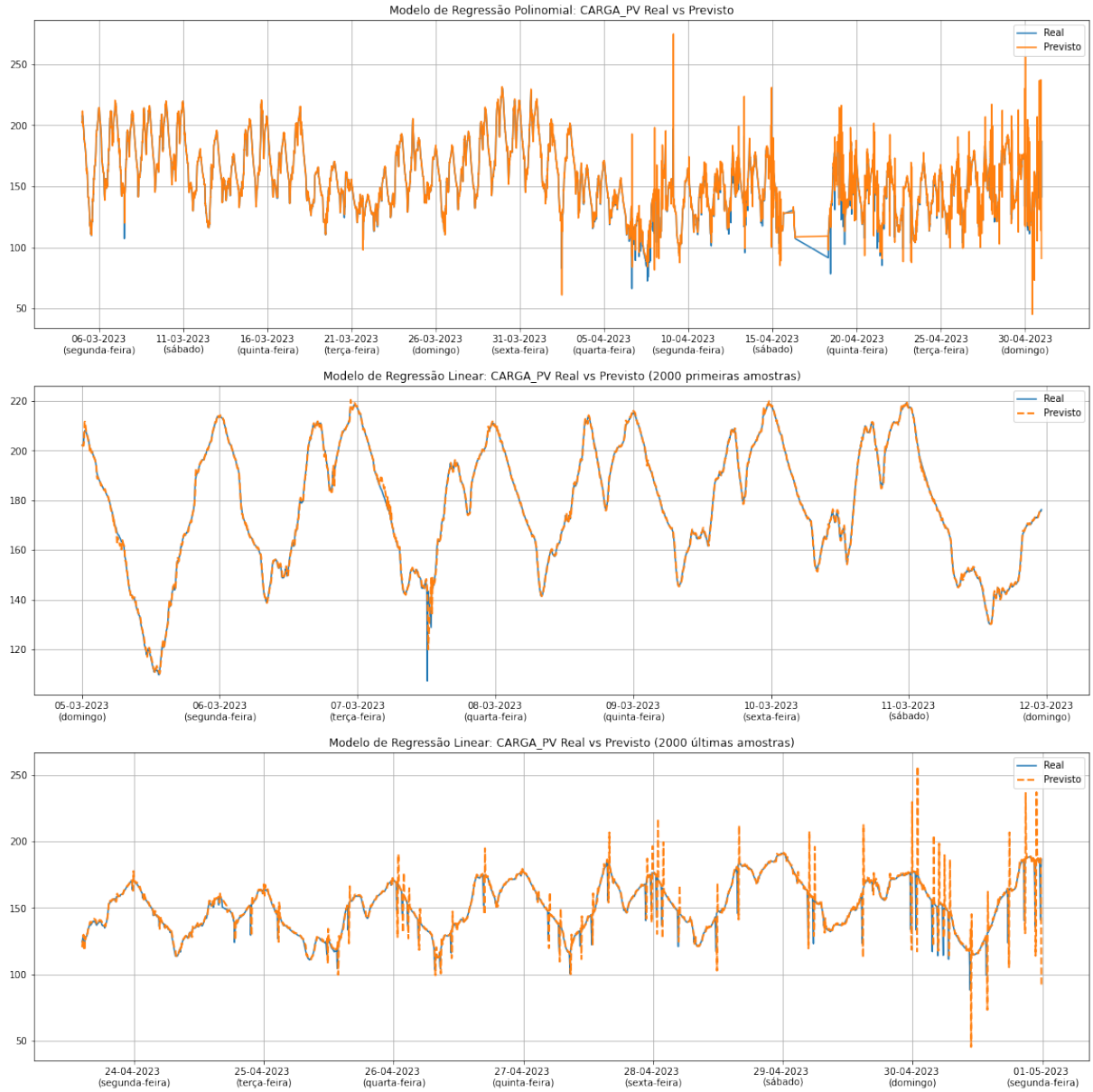
Neste tópico, o desempenho dos modelos é avaliado graficamente, por meio da sobreposição das curvas de predição com os dados reais de medição, integrantes do conjunto de dados de teste. Para a melhor visualização dos resultados, serão apresentados no mesmo quadro as curvas completas de predição e de teste, bem como um quadro com as 2000 primeiras amostras e outro com as 2000 últimas, a fim de destacar os detalhes da generalização de cada modelo.

Figura 33 – Comparação entre dados reais de carregamento no transformador e predição do modelo de Regressão Linear



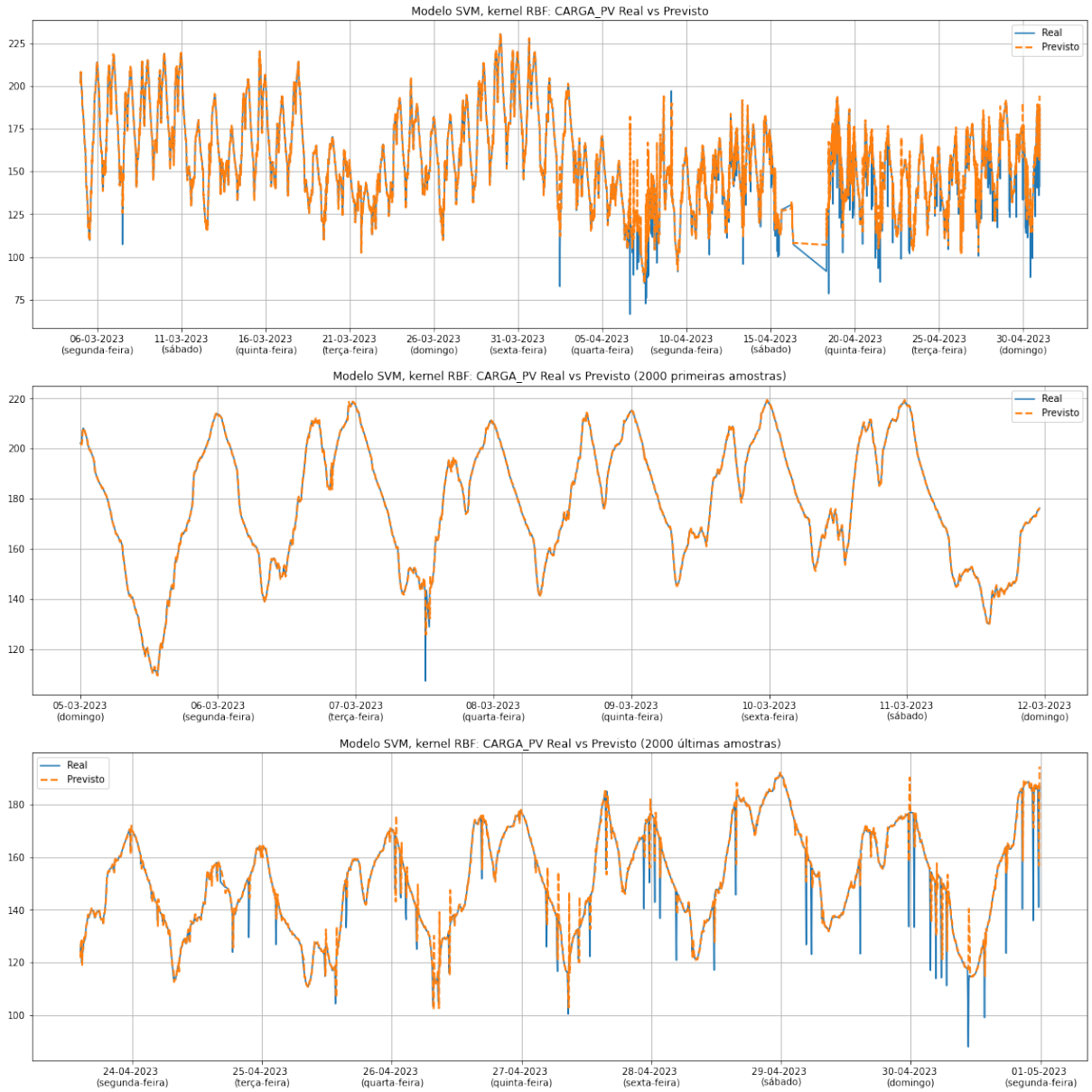
Fonte: do próprio autor.

Figura 34 – Comparação entre dados reais de carregamento no transformador e predição do modelo de Regressão Polinomial



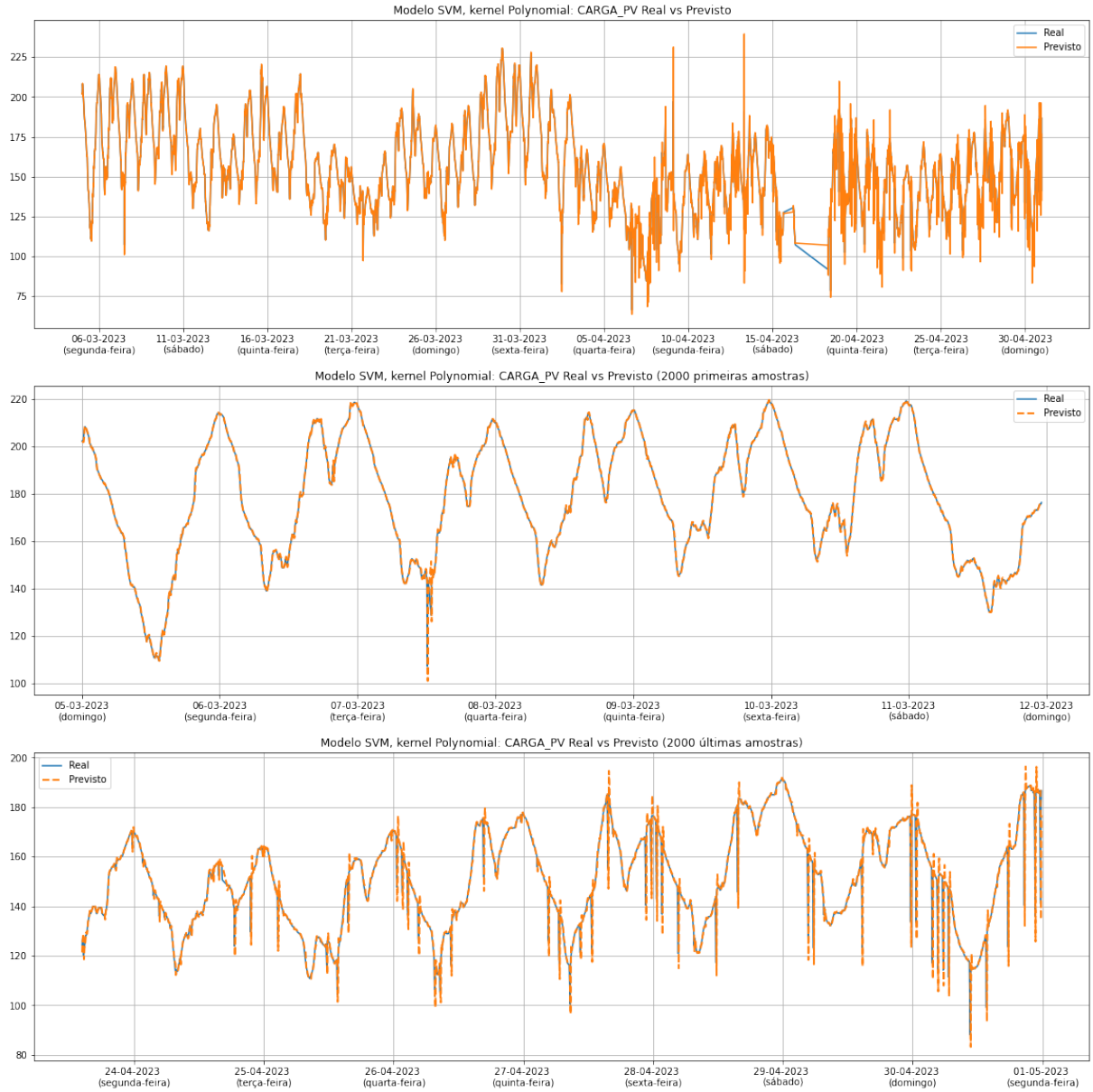
Fonte: do próprio autor.

Figura 35 – Comparação entre dados reais de carregamento no transformador e predição do modelo SVM, *kernel* RBF



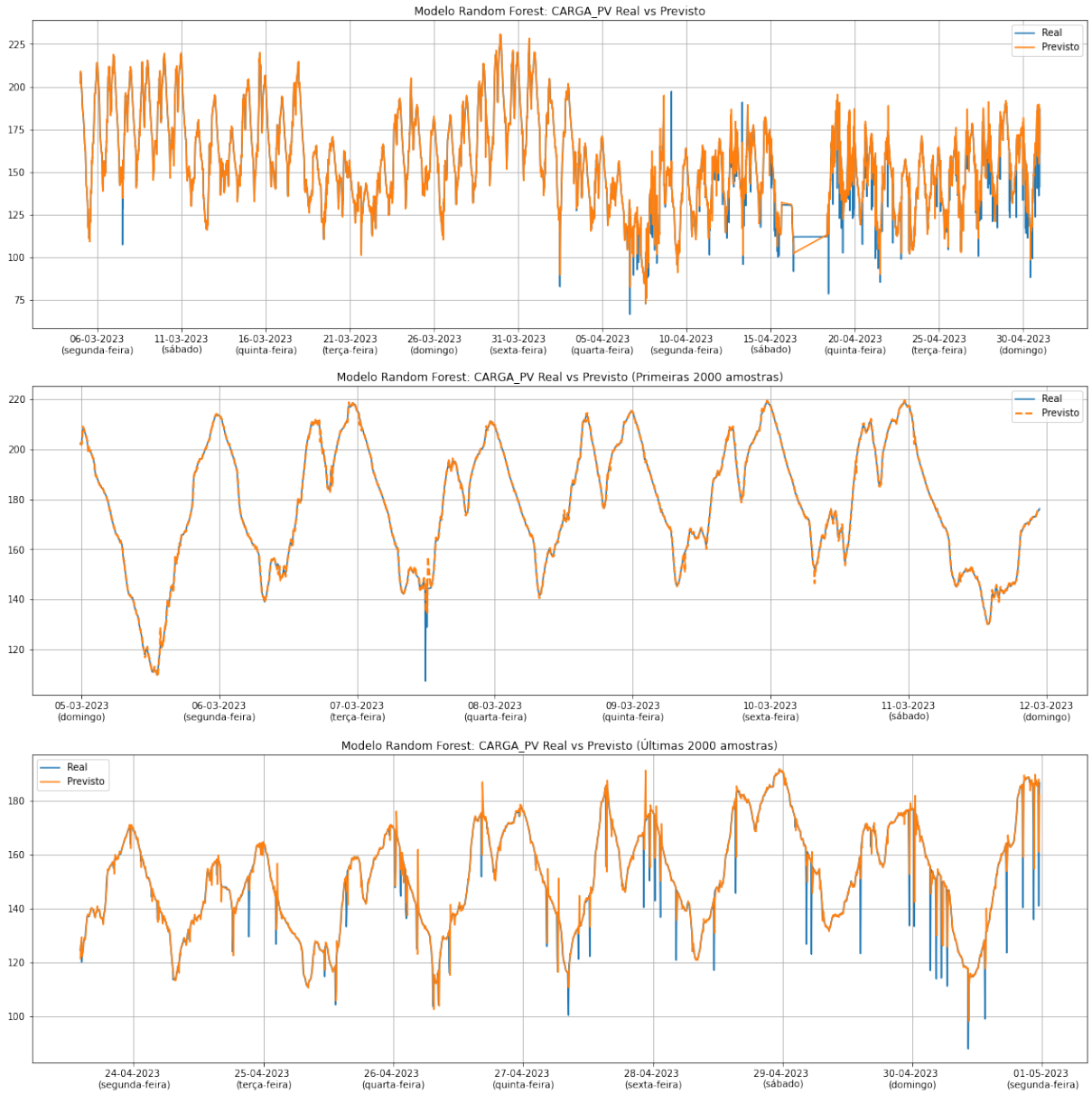
Fonte: do próprio autor.

Figura 36 – Comparação entre dados reais de carregamento no transformador e predição do modelo SVM, *kernel* polynomial



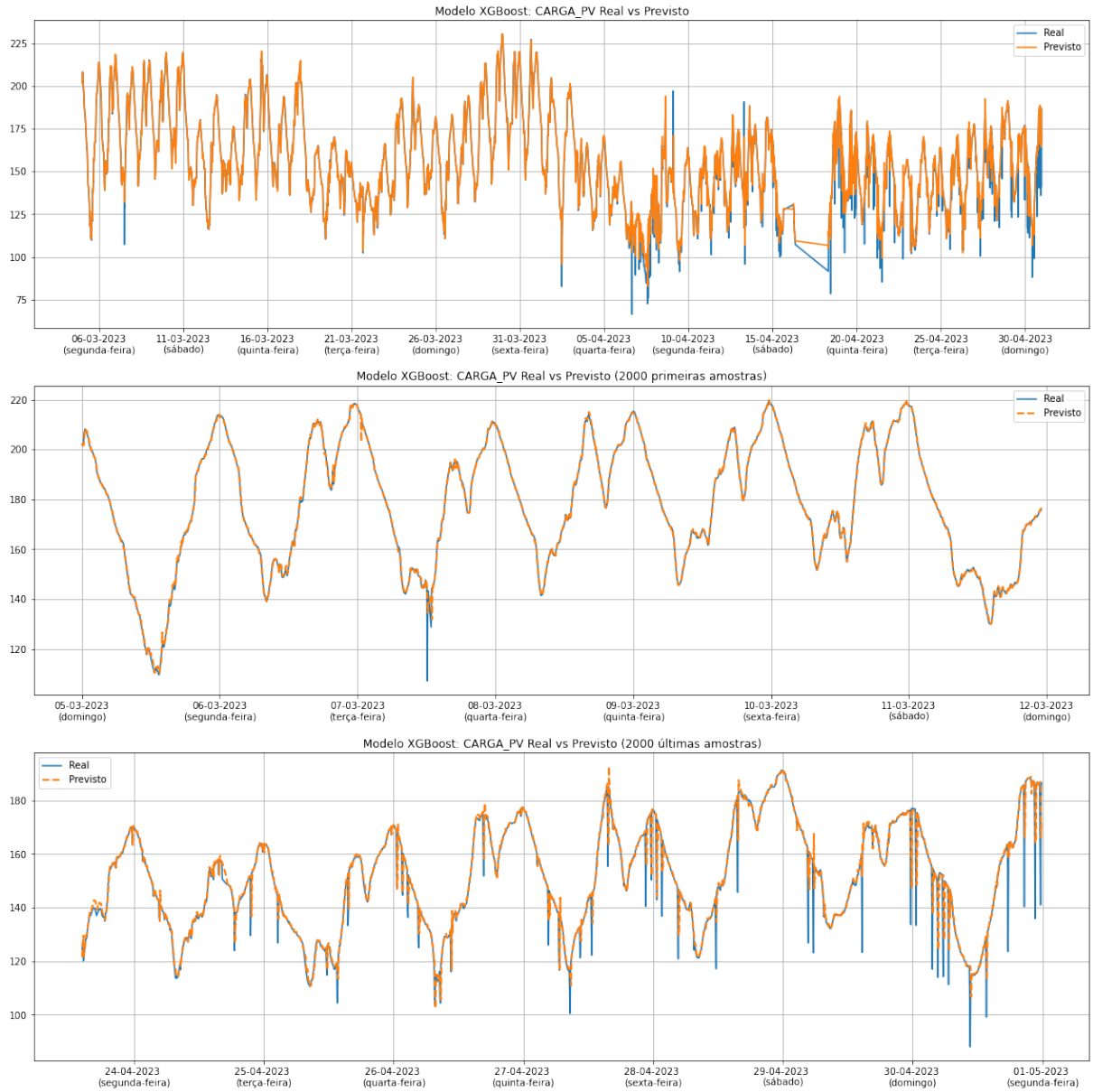
Fonte: do próprio autor.

Figura 37 – Comparação entre dados reais de carregamento no transformador e predição do modelo *Random Forest*



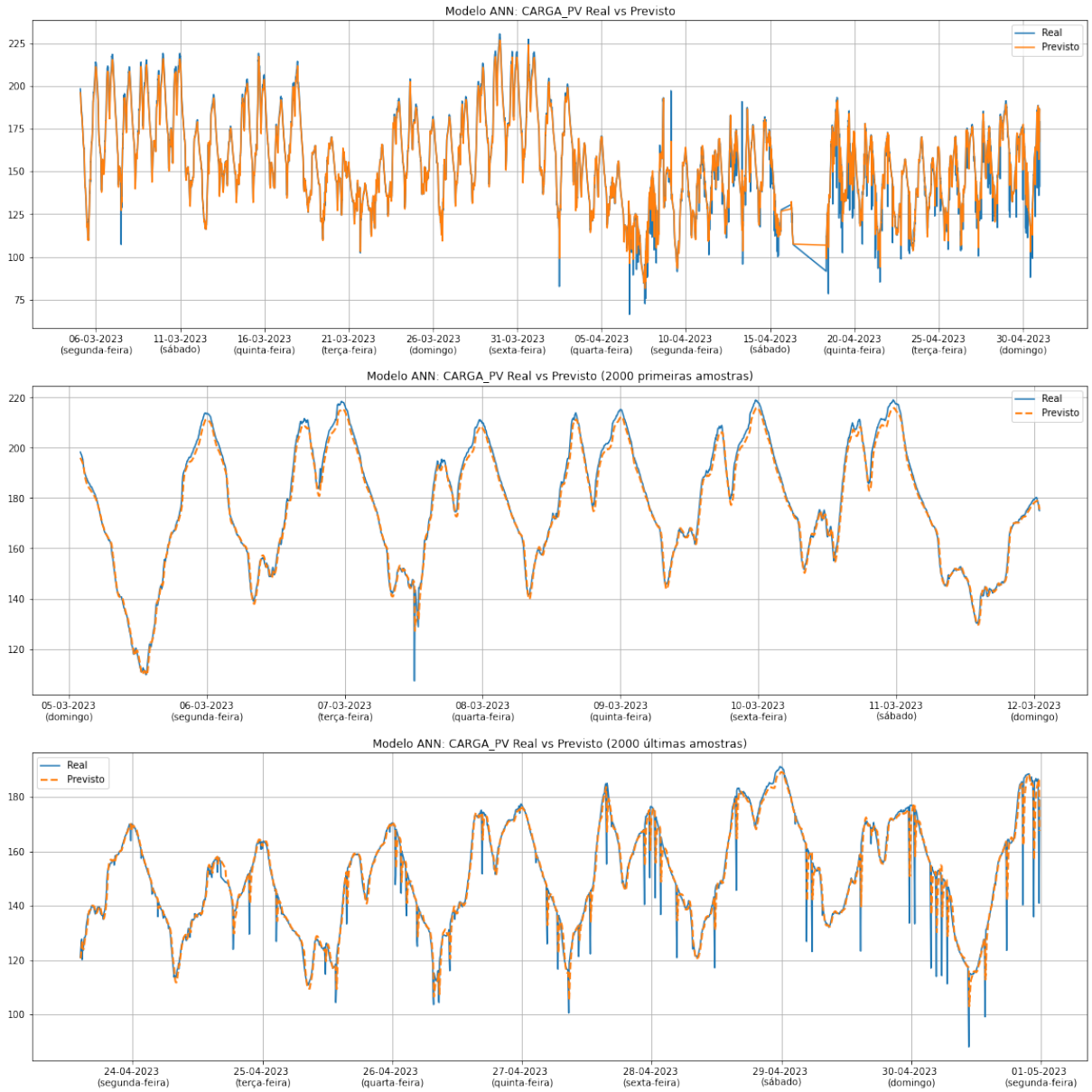
Fonte: do próprio autor.

Figura 38 – Comparação entre dados reais de carregamento no transformador e predição do modelo XGBoost



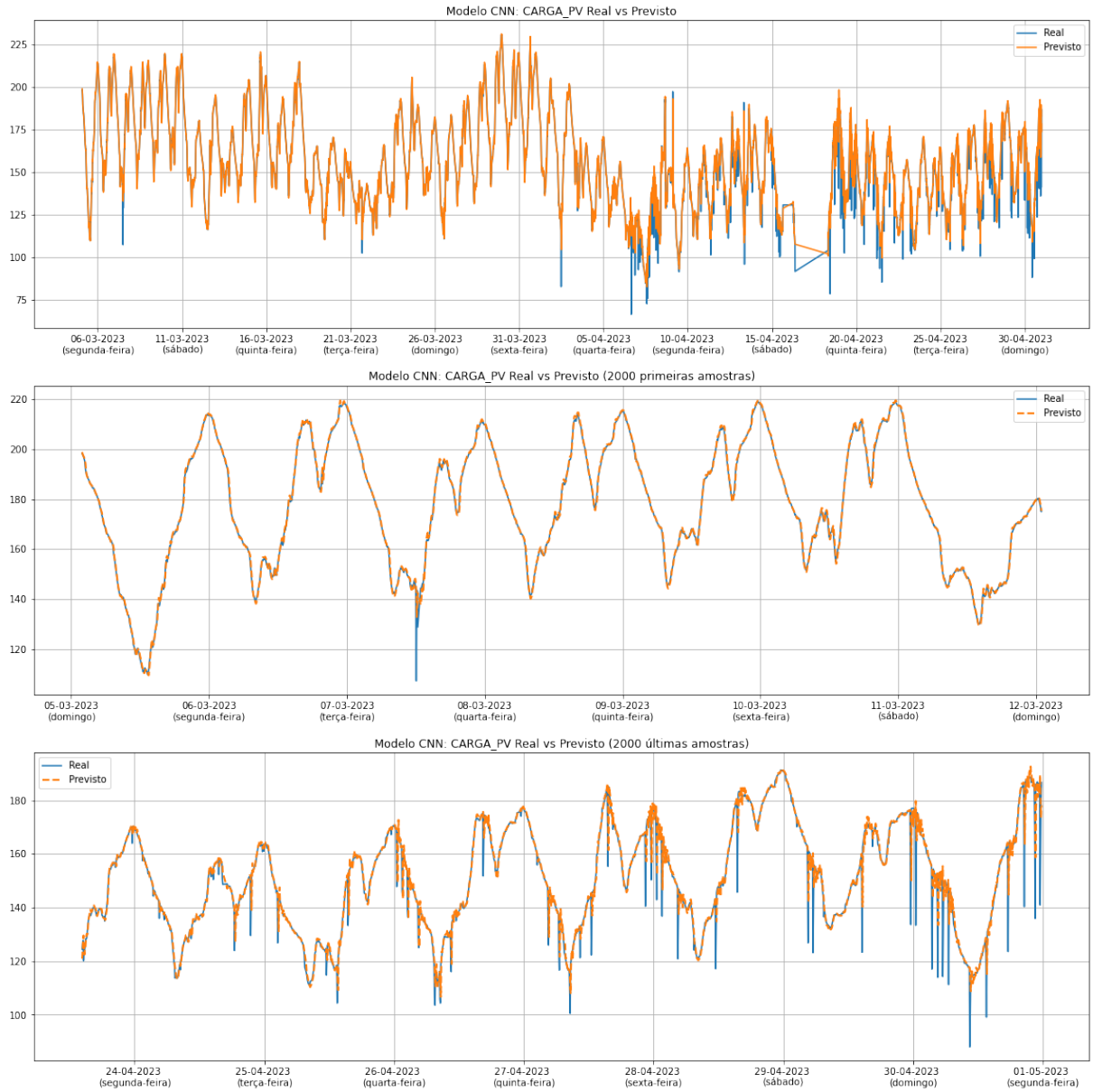
Fonte: do próprio autor.

Figura 39 – Comparação entre dados reais de carregamento no transformador e predição do modelo ANN



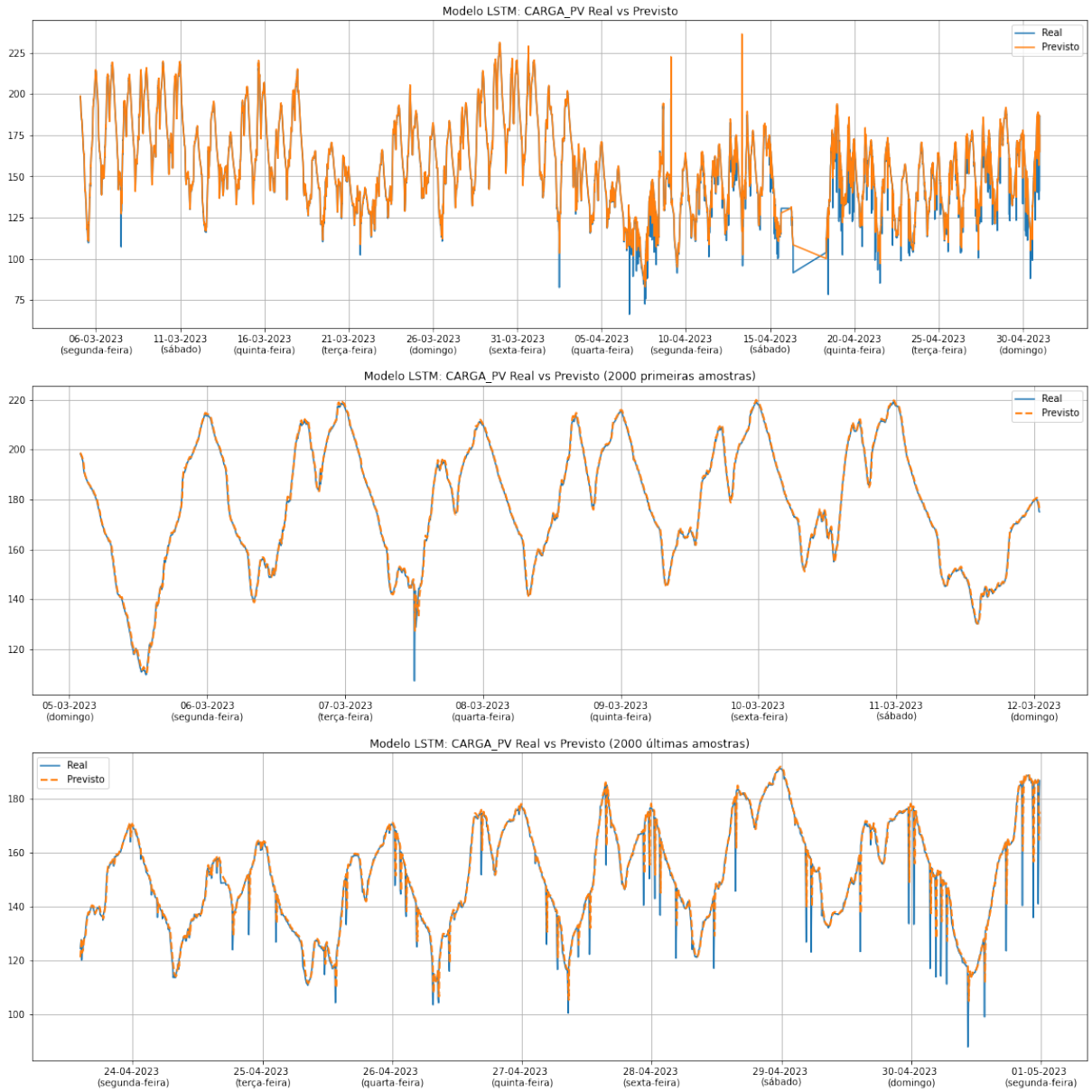
Fonte: do próprio autor.

Figura 40 – Comparação entre dados reais de carregamento no transformador e predição do modelo CNN



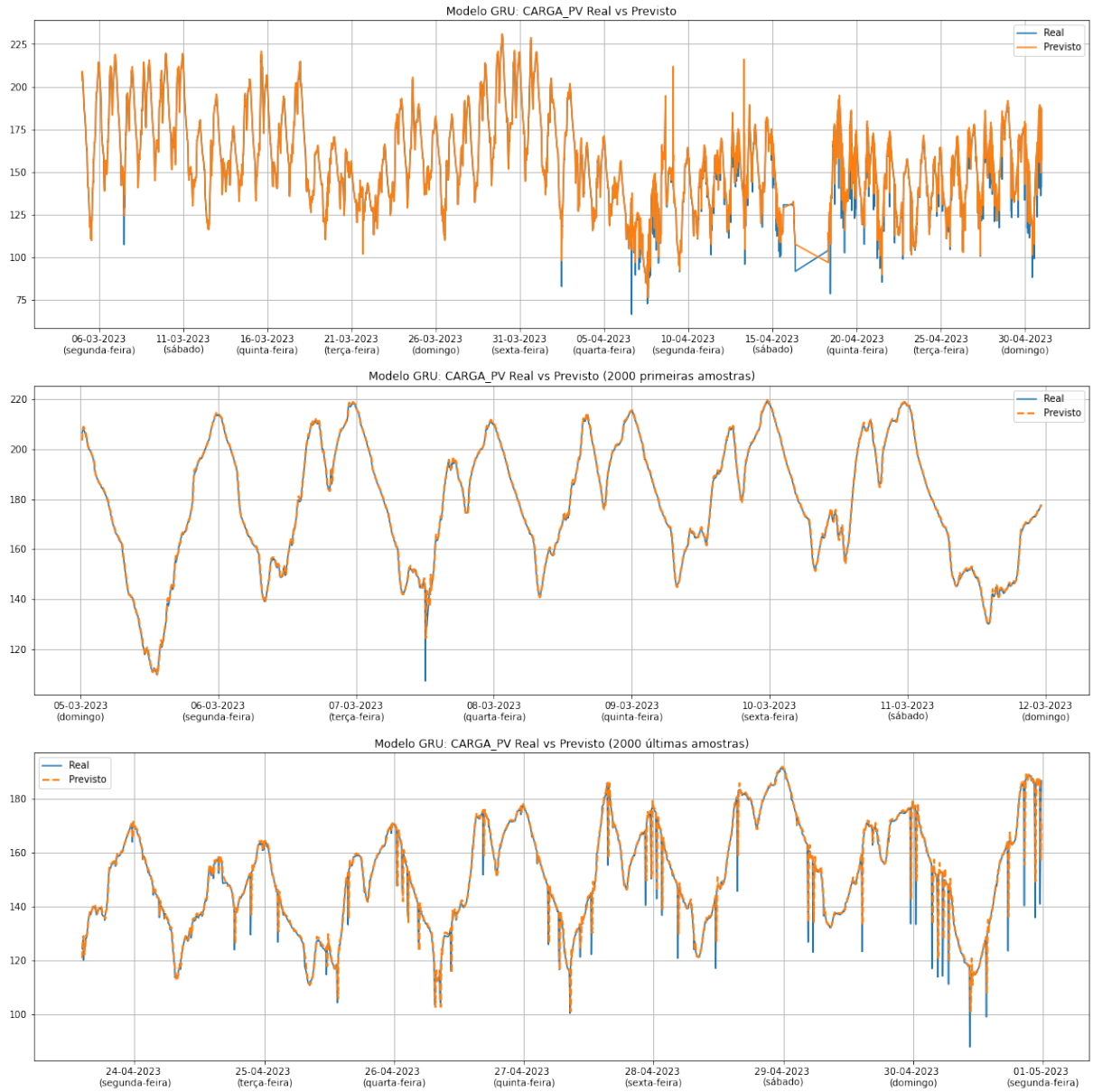
Fonte: do próprio autor.

Figura 41 – Comparação entre dados reais de carregamento no transformador e predição do modelo LSTM



Fonte: do próprio autor.

Figura 42 – Comparação entre dados reais de carregamento no transformador e predição do modelo GRU



Fonte: do próprio autor.

Todas as figuras anteriores demonstram aquilo que a Tabela 19 já havia apresentado: Todos os modelos adotados, implementados conforme as configurações apresentadas na seção 4.5, foram plenamente capazes de se ajustar aos conjunto de dados de treinamento e entregar excelentes resultados de generalização, sendo capazes de prever a carga futura do conjunto de teste com bastante precisão. Conforme observação dos gráficos, a diferença perceptível entre as curvas dos modelos se dá, basicamente, em como cada um lida com váriações bruscas de carga (ou *spikes*) que, para efeito de planejamento elétrico da operação do sistema de potência, são dispensáveis.

Assim, pode-se afirmar com segurança, que todos os modelos adotados neste

trabalho, sob o aspecto de precisão dos resultados, têm a mesma capacidade de prever a carga futura. O critério para a escolha do melhor modelo, portanto, passa a ser a complexidade de implementação e o tempo de processamento de cada modelo.

Um fator determinante para obter estes resultados foi a utilização do recurso de *look-back* e de janelas deslizantes, conforme será demonstrado por meio das métricas de validação e da análise gráfica no item 5.7.4, juntamente com a discussão da influência das demais *features* do conjunto de dados na predição da variável-alvo 'Carga_PV'.

5.7.4 Influência das Features e do recurso de *Look-Back* ou Janelas Deslizantes nos resultados de predição de carga

Neste tópico, serão apresentados os efeitos das *features* na projeção de dados futuros para a variável-alvo em estudo. Tal demonstração, conforme indicado na seção 4.6, será conduzida por meio de funções tais como o *Ordinary Least Squares* e o *Features Importances*, assim como utilizando métricas de validação e análise gráfica. Para efeito comparativo, os mesmos modelos serão aplicados ao conjunto de dados, porém, desta vez sem os atributos climáticos e medindo o novo desempenho dos modelos. Exemplos comparativos de gráficos de predição com e sem a utilização destes atributos também serão apresentados.

Além disso, será discutida a significância do *look-back* e das janelas deslizantes no contexto do modelo apropriado. Para tal, serão apresentadas métricas de validação e gráficos gerados com a implementação da arquitetura de modelo idêntica em um conjunto de treinamento desprovido dos atributos derivados do *look-back* e das janelas deslizantes, confrontando-os com os resultados obtidos a partir do conjunto de dados completo.

Objetiva-se, portanto, destacar a relevância dos atributos na predição, por isso foram selecionados apenas alguns modelos para exemplificar a aplicação destas técnicas, evitando redundância e a repetição desnecessária das mesmas técnicas aplicada a todos os modelos.

5.7.4.1 Influência dos Atributos Climáticos

A Figura 25 já havia dado um indicativo da influência (ou correlação) dos atributos climáticos sobre a variável-alvo **'CARGA_PV'**. De acordo com aquela matriz de correlação, os atributos com maiores influências sobre a Carga são a 'Pressão Atmosférica', com -0.45 e indicando correlação negativa moderada, e a 'Temperatura', indicando uma correlação positiva fraca. Inicialmente, este teste sugere que estas características oferecem pouca influência no processo de predição da carga, dada a moderada à baixa correlação entre elas.

Para uma avaliação mais robusta, foi utilizada a ferramenta *Ordinary Least Squares* - OLS, da biblioteca *statsmodels* em duas etapas: Aplicando no conjunto de dados completo e depois, num conjunto de dados sem os atributos climáticos. Paralelamente, esses dois

conjuntos de dados foram aplicados aos dois modelos de regressão adotados neste trabalho, visando avaliar o desempenho de ambos com e sem as características de clima.

A Figura 43 é o resultado do teste de OLS sobre o conjunto completo enquanto a Figura 44 é o mesmo conjunto de dados, sem as variáveis climáticas:

Figura 43 – OLS do conjunto completo

OLS Regression Results

Dep. Variable:	CARGA_PV	R-squared:	0.993
Model:	OLS	Adj. R-squared:	0.993
Method:	Least Squares	F-statistic:	1.098e+06
Date:	sáb, 05 ago 2023	Prob (F-statistic):	0.00
Time:	15:40:06	Log-Likelihood:	-1.8281e+05
No. Observations:	75988	AIC:	3.656e+05
Df Residuals:	75977	BIC:	3.657e+05
Df Model:	10		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	64.1329	0.167	383.734	0.000	63.805	64.461
tipo_dia	0.0921	0.021	4.304	0.000	0.050	0.134
Precip	-0.8897	0.222	-4.005	0.000	-1.325	-0.454
Umid. Relat. AR (%)	0.5044	0.105	4.789	0.000	0.298	0.711
Temperatura do ar (C)	3.4495	0.152	22.768	0.000	3.153	3.746
Pressão Atm.	0.7062	0.090	7.886	0.000	0.531	0.882
hora_flt	1.2051	0.036	33.340	0.000	1.134	1.276
data_flt	-0.2354	0.048	-4.903	0.000	-0.329	-0.141
CARGA_PV_lag_1	165.3476	0.790	209.277	0.000	163.799	166.896
CARGA_PV_lag_2	36.3343	0.983	36.950	0.000	34.407	38.262
CARGA_PV_lag_3	14.8577	0.789	18.822	0.000	13.311	16.405

Omnibus:	80814.428	Durbin-Watson:	1.992
Prob(Omnibus):	0.000	Jarque-Bera (JB):	223654416.539
Skew:	-4.167	Prob(JB):	0.00
Kurtosis:	268.649	Cond. No.	246.

Fonte: do próprio autor.

Observando os dois resultados e a diferença entre ambos, conclui-se que:

i. No conjunto de dados completo, as variáveis mais significativas são aquelas oriundas do recurso de *look-back* com observação de até [*t-3*], com influência muito mais forte na previsão da variável-alvo **'Carga_PV'** que todos atributos climáticos;

ii. As variáveis climáticas, segundo os critérios do teste de OLS, mostraram algum grau de significância, embora sua significância seja muito inferior que àquelas decorrentes de técnica de observação de amostras passadas.

iii. O atributo 'tipo_dia', embora significativo, agrega muito pouco à qualidade da previsão da carga.

Figura 44 – OLS do conjunto sem atributos climáticos

OLS Regression Results

Dep. Variable:	CARGA_PV	R-squared:	0.993
Model:	OLS	Adj. R-squared:	0.993
Method:	Least Squares	F-statistic:	1.794e+06
Date:	sáb, 05 ago 2023	Prob (F-statistic):	0.00
Time:	15:53:19	Log-Likelihood:	-1.8358e+05
No. Observations:	75988	AIC:	3.672e+05
Df Residuals:	75981	BIC:	3.672e+05
Df Model:	6		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	66.2839	0.048	1367.165	0.000	66.189	66.379
tipo_dia	0.0643	0.022	2.991	0.003	0.022	0.106
hora_flt	1.5177	0.035	43.400	0.000	1.449	1.586
data_flt	-0.2021	0.039	-5.191	0.000	-0.278	-0.126
CARGA_PV_lag_1	169.5001	0.791	214.304	0.000	167.950	171.050
CARGA_PV_lag_2	36.4384	0.993	36.686	0.000	34.492	38.385
CARGA_PV_lag_3	11.0502	0.790	13.979	0.000	9.501	12.600

Omnibus:	75968.038	Durbin-Watson:	1.993
Prob(Omnibus):	0.000	Jarque-Bera (JB):	213555899.901
Skew:	-3.674	Prob(JB):	0.00
Kurtosis:	262.606	Cond. No.	209.

Fonte: do próprio autor.

iv. Ambos os resultados apresentaram a mesma métrica R2 e R2-ajustado, indicando que a retirada das variáveis climáticas em nada influenciou a qualidade do ajuste do modelo.

Em suma, a observação entre a comparação entre a resposta ao OLS dos dois conjuntos, com e sem atributos climáticos, demonstra que no modelo com o conjunto de dados completo, as variáveis *lag* desempenham um papel dominante na previsão da carga. A influência de outras variáveis, incluindo as climáticas, é relativamente pequena em comparação. Essa observação indica que a dinâmica temporal da carga, representada pelos valores de lag, é um fator crítico na modelagem dessa série temporal, enquanto os atributos climáticos e outros têm uma contribuição menor ou perto de nula.

Esta conclusão também pode ser verificada por meio do resultado do teste *Features Importances*, utilizado nos modelos baseados em árvores de decisão. A Tabela 20 apresenta a influência das *features* do conjunto de dados no processo decisório da variável-alvo:

Novamente, é confirmada a indicação que a observação temporal da dinâmica da carga é mais importante que a observação climática para a determinação dos dados futuros,

Tabela 20 – Importância das características para os modelos *Random Forest* e XGBoost.

Feature	*Random Forest*	XGBoost
CARGA_PV_lag_1	0.98222	0.96782
CARGA_PV_lag_2	0.01221	0.01722
CARGA_PV_lag_3	0.00382	0.00848
tipo_dia	0.00003	0.00068
Precip	0.00012	0.00068
Umid. Relat. AR (%)	0.00022	0.00067
Temperatura do ar	0.00026	0.00076
Pressão Atm.	0.00034	0.00079
hora_flt	0.00043	0.00126
data_flt	0.00036	0.00059

Fonte: do próprio autor.

Tabela 21 – Comparação de desempenho entre os modelos com e sem atributos climáticos

Modelo	**Parâmetro**	**Completo**	**sem Atributos Climáticos**
Regressão Linear	R2	0.98	0.982
	MSE	12.21	12.39
	Tempo de treinamento	40.9 ms	8.98 ms
	Tempo de previsão	2.99 ms	2.99 ms
Regressão Polinomial	R2	0.964	0.965
	MSE	25.506	24.512
	Tempo de treinamento	860 ms	240 ms
	Tempo de previsão	47.1 ms	24.8 ms
SVM rbf	R2	0.981	0.982
	MSE	13.457	12.440
	Tempo de treinamento	7 min 57s	4min 27s
	Tempo de previsão	1min 38s	1min 42s
SVM poly	R2	0.976	0.976
	MSE	16.770	17.059
	Tempo de treinamento	4min 40s	5min 19s
	Tempo de previsão	25.4 s	22.4 s
Random Forest	R2	0.985	0.984
	MSE	0.00023	10.941
	Tempo de treinamento	42.6 s	37.9 s
	Tempo de previsão	354 ms	337 ms
XGBoost	R2	0.982	0.983
	MSE	12.443	11.789
	Tempo de treinamento	6.44 s	5.78 s
	Tempo de previsão	20.6 ms	21.9 ms

Fonte: do próprio autor.

não só nos modelos de regressão, mas também nos modelos de baseados em *machine learning* - para esta aplicação e conjunto de dados específico.

Em termos práticos, ao submeter o conjunto de dados sem atributos climáticos aos mesmos modelos adotados para realizar a previsão da carga, os seguintes resultados de R2, MSE e tempo de processamento foram obtidos e compilados na Tabela 21.

Diante desta comparação, evidencia-se o fato de que a presença das variáveis climáticas não é, na maioria dos casos, crucial para a qualidade da previsão dos dados futuros, e que em alguns modelos, a ausência até melhora atributos de qualidade e de tempo de processamento. Este comportamento deve-se ao fato de que o modelo se ajustou melhor a um conjunto de dados mais simples, em que a complexidade reduzida pode levar

a menos ruído e a uma maior eficiência na modelagem. Essa simplicidade permite que o modelo se concentre nos atributos que têm uma correlação mais direta com a variável-alvo, aumentando potencialmente tanto a velocidade de treinamento quanto a precisão na previsão.

Esta observação impacta diretamente na concepção final de um preditor de carga para ser colocado em produção, considerando que a obtenção e o tratamento dos dados climáticos não é uma tarefa simples. Uma vez comprovado que, para esta aplicação e conjunto de dados específica, esse tipo de variável pouco influencia a qualidade da previsão - e sua ausência pode ainda tornar o preditor mais rápido, como na maioria dos modelos da Tabela 21.

5.7.4.2 Influência dos atributos de observação temporal: *look-back* e janelas deslizantes

Analogamente, de forma a complementar a análise anterior, foram realizadas novas previsões com os mesmos modelos - agora mantendo as variáveis climáticas e suprimindo aquelas decorrentes de observações temporais da variável-alvo anteriores, como as 'CARGA_PV_lag_n' - com o objetivo de verificar o quanto estas variáveis são importantes para a previsão e se, sem elas, é possível prever a carga futura em algum intervalo de tempo.

Para o novo OLS para o conjunto de dados com todas as variáveis climáticas mas sem o artifício de *look-back* é apresentado na Figura 45

Percebe-se a considerável diminuição do parâmetro de R2 indicando a perda da qualidade do ajuste do modelo ao conjunto de dados e a consequente diminuição da qualidade do resultado da previsão. Variáveis, outrora com baixo coeficiente, ganham maior relevância, como a 'Temperatura do ar', a 'Umidade relativa do ar' e até o 'tipo do dia' que, conforme expectativa anterior, esperava-se que realmente tivessem influência sobre a determinação da carga futura.

Entretanto, conforme demonstra a Tabela 22, análoga à Tabela 21, somente estes atributos sem apoio das variáveis de dinâmica temporal não são suficientes para realizar a previsão de carga de forma adequada.

A Tabela 22 novamente demonstra que o recurso de *look-back* é crucial para a qualidade da previsão nos modelos testados. Sem esse recurso, a maioria dos modelos experimenta uma queda significativa na qualidade de ajuste, conforme evidenciado por R2 negativo ou valores MSE mais altos.

Para entender o que significa os parâmetros de R2 e MSE tão longes do valor ideal de conjunto completo, as Figuras 46, 47 e 48 ilustram a saída gráfica dos modelos de regressão polinomial, SVM polinomial e XGBoost, respectivamente.

Comprova-se então que, para esta aplicação, conjunto de dados específico e para os

Figura 45 – OLS do conjunto sem *look-back*

OLS Regression Results

Dep. Variable:	CARGA_PV	R-squared:	0.431
Model:	OLS	Adj. R-squared:	0.431
Method:	Least Squares	F-statistic:	8210.
Date:	ter, 08 ago 2023	Prob (F-statistic):	0.00
Time:	02:24:49	Log-Likelihood:	-3.5066e+05
No. Observations:	75991	AIC:	7.013e+05
Df Residuals:	75983	BIC:	7.014e+05
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	126.8644	1.500	84.576	0.000	123.924	129.804
tipo_dia	11.8384	0.190	62.319	0.000	11.466	12.211
Precip	13.4672	2.021	6.662	0.000	9.505	17.429
Umid. Relat. AR (%)	73.1747	0.921	79.428	0.000	71.369	74.980
Temperatura do ar (C)	113.0573	1.316	85.886	0.000	110.477	115.637
Pressão Atm.	-87.0977	0.747	-116.567	0.000	-88.562	-85.633
hora_flt	9.0021	0.325	27.722	0.000	8.366	9.639
data_flt	-58.0102	0.383	-151.643	0.000	-58.760	-57.260

Omnibus:	837.649	Durbin-Watson:	0.017
Prob(Omnibus):	0.000	Jarque-Bera (JB):	864.563
Skew:	0.260	Prob(JB):	1.83e-188
Kurtosis:	2.948	Cond. No.	44.0

Fonte: do próprio autor.

modelos selecionados, a utilização de atributos de observação temporal no conjunto de dados é fator decisivo de sucesso nos previsores de carregamento ao qual o trabalho se propõe.

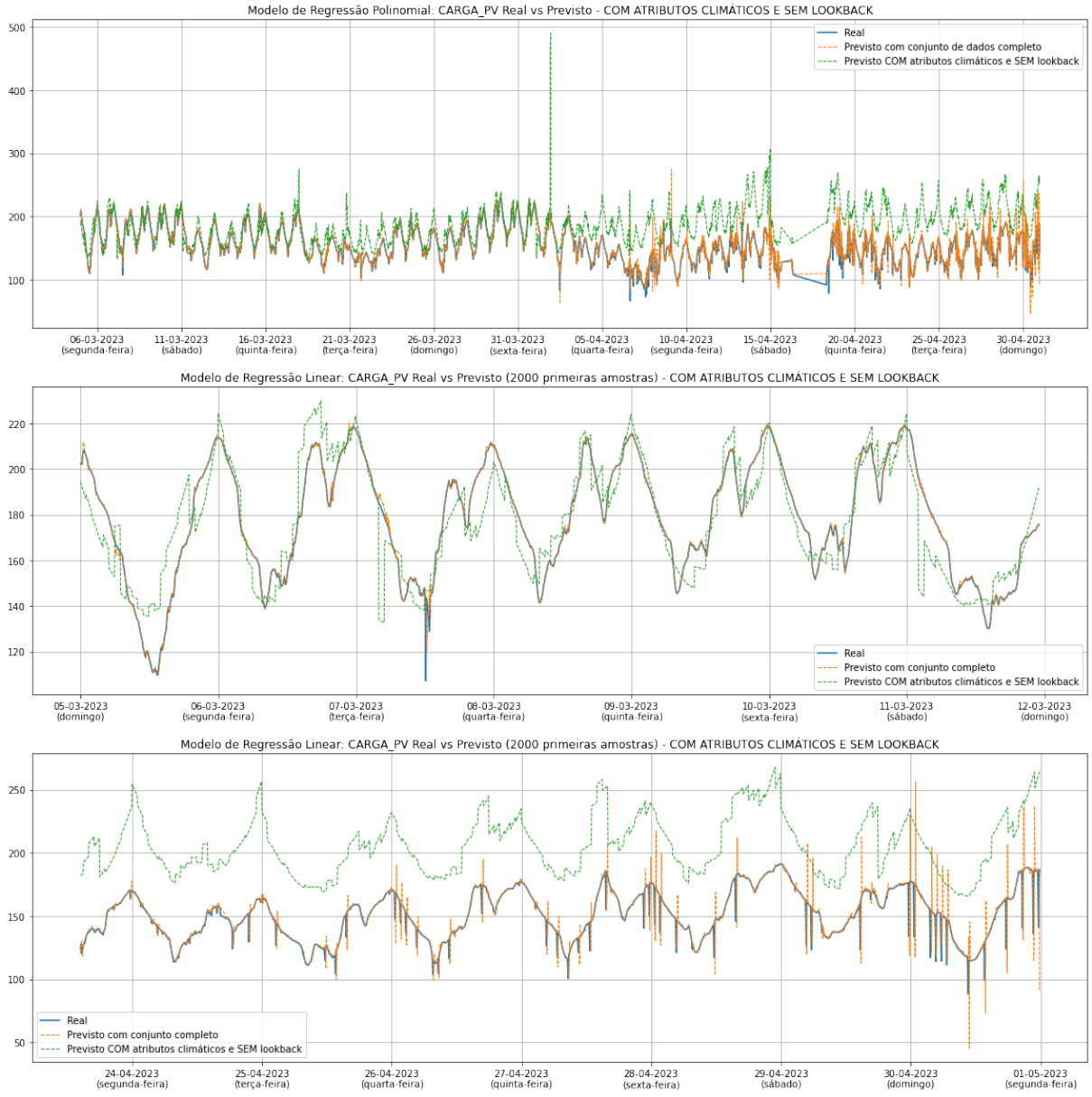
No caso dos modelos baseados em Redes Neurais, optou-se de início não utilizar os atributos climáticos no conjunto de dados. Os resultados apresentados anteriormente, na Tabela 19, já não contemplavam estas características.

Para avaliar a influência das janelas deslizantes sobre os modelos de redes neurais, processo análogo realizado aos atributos derivados de *look-back* foi realizado: Conjunto de dados sem característica de observação temporal da janela deslizante foi aplicado a cada modelo de rede neural adotado, medida as métricas de avaliação e os tempos de treinamento e teste e, finalmente, comparado com a execução original com conjunto completo. Os resultados são apresentados na Tabela 23. Temos que:

i. O desempenho do modelo ANN reduziu um pouco, sob aspecto das métricas de avaliação R2 e MSE, ao custo de uma melhora dos tempos de processamento quase que insignificante. Neste caso, a utilização das janelas deslizantes é justificável, caso a busca pela maior precisão da previsão dos dados seja um critério inegociável;

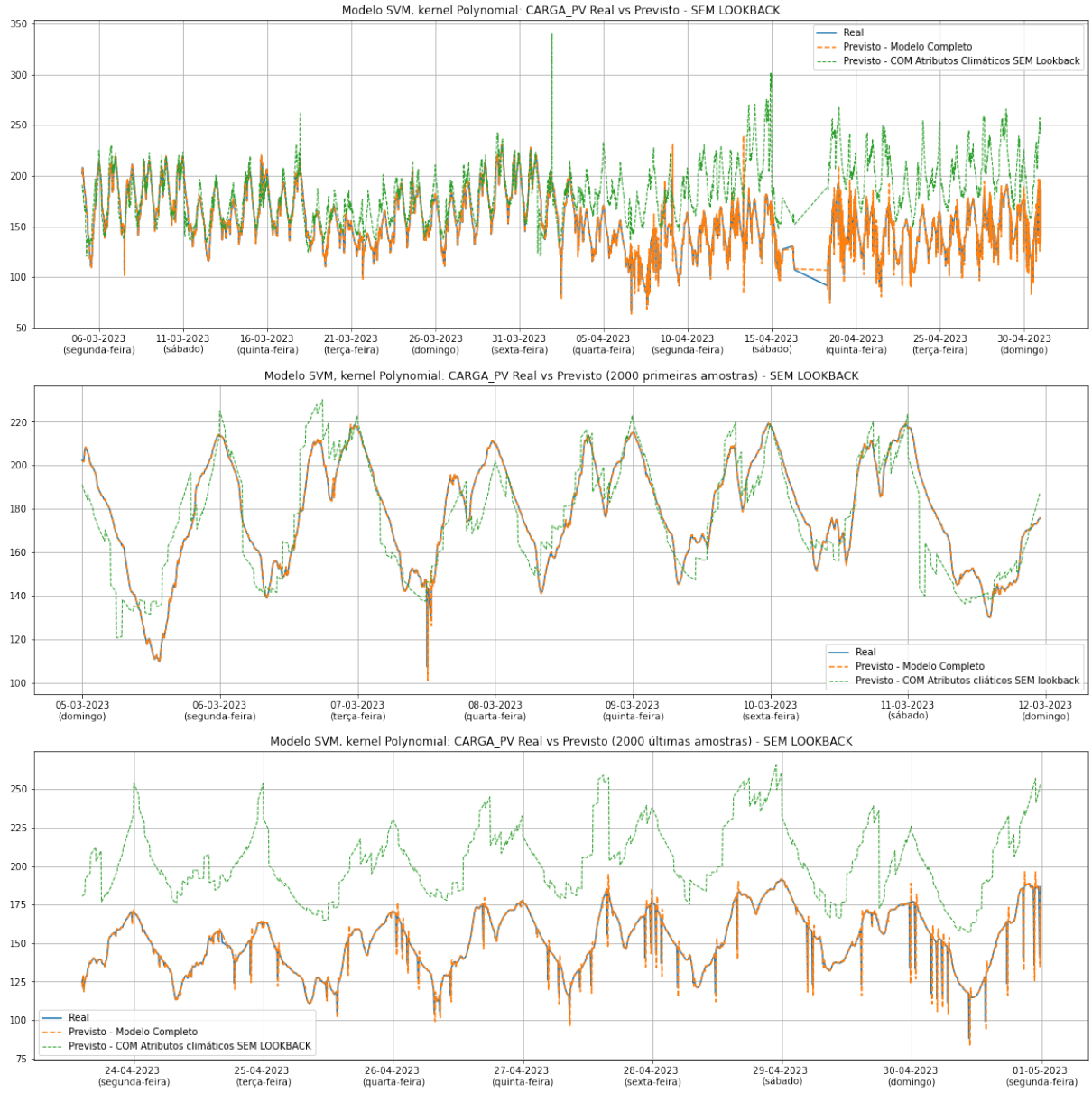
ii. No caso da CNN, há uma perda quase que imperceptível no desempenho do modelo ao custo de uma boa melhoria nos tempos de treinamento e teste, sendo muito

Figura 46 – Previsão de carga via regressor polinomial com conjunto de dados sem *look-back*



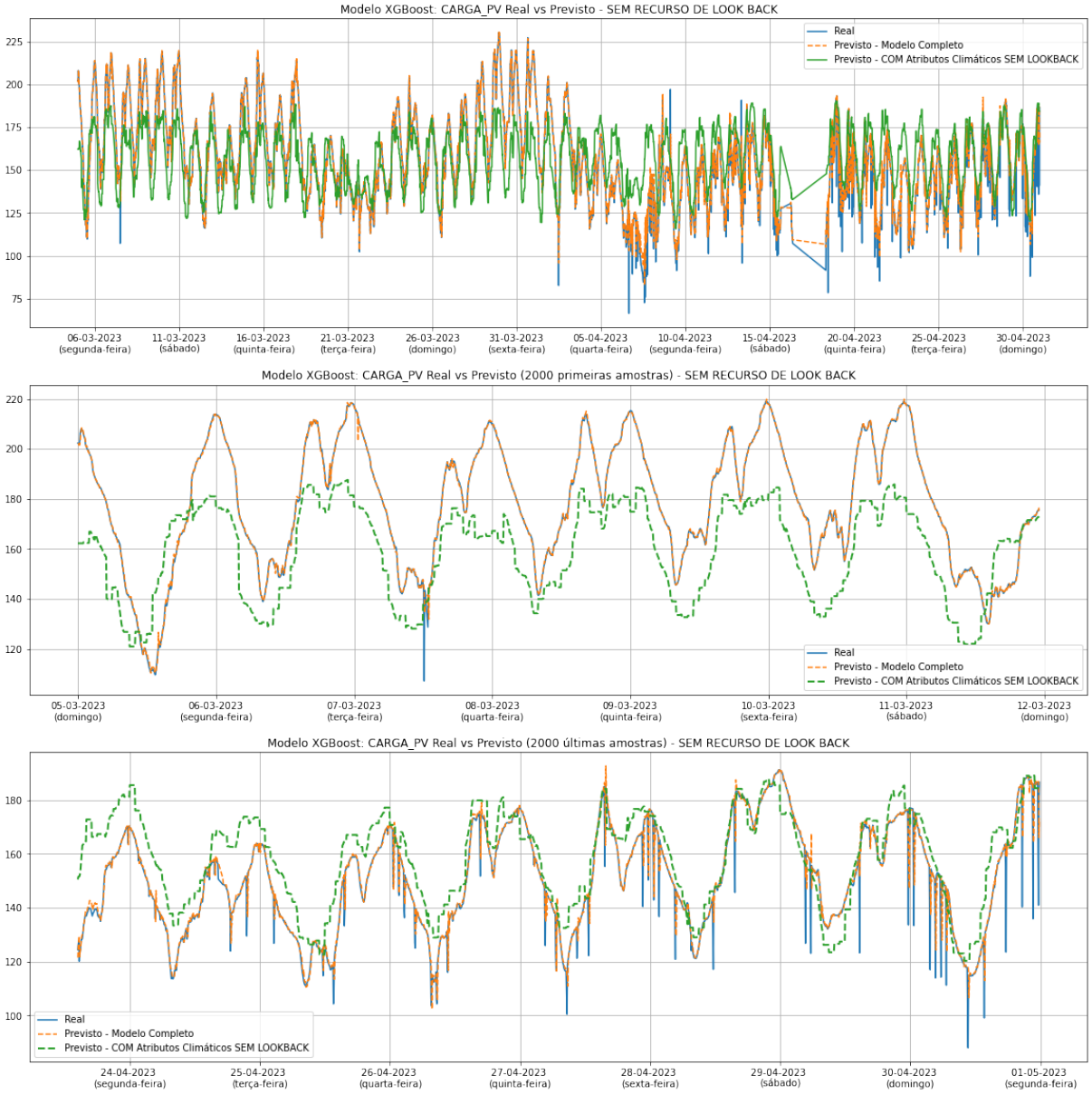
Fonte: do próprio autor.

Figura 47 – Previsão de carga via SVM polinomial com conjunto de dados sem *look-back*



Fonte: do próprio autor.

Figura 48 – Previsão de carga via regressor polinomial com conjunto de dados sem *look-back*



Fonte: do próprio autor.

Tabela 22 – Comparação dos modelos com e sem o recurso de *look-back*.

Modelo	**Parâmetro**	**Conjunto de Dados**	
		Completo	**sem *look-back***
Regressão Linear	R2	0.98	0.079
	MSE	12.21	645.18
	Tempo de treinamento	40.9 ms	9.98 ms
	Tempo de previsão	2.99 ms	2.99 ms
Regressão Polinomial	R2	0.964	1607.078
	MSE	25.506	-1.294
	Tempo de treinamento	860 ms	420 ms
	Tempo de previsão	47.1 ms	32.3 ms
SVM rbf	R2	0.981	0.082
	MSE	13.457	643.041
	Tempo de treinamento	7 min 57s	5min 10s
	Tempo de previsão	1min 38s	2min 3s
SVM poly	R2	0.976	-1.122
	MSE	16.770	1486.511
	Tempo de treinamento	4min 40s	14min 14s
	Tempo de previsão	25.4 s	29.5 s
Random Forest	R2	0.985	-0.484
	MSE	0.00023	1039.402
	Tempo de treinamento	42.6 s	14.8 s
	Tempo de previsão	354 ms	154 ms
XGBoost	R2	0.982	0.425
	MSE	12.443	403.038
	Tempo de treinamento	6.44 s	8.27 s
	Tempo de previsão	20.6 ms	32 ms

Fonte: do próprio autor.

plausível a escolha pelo não uso do recurso de janelas deslizantes nesta aplicação;

iii. No caso das LSTM e GRU, a não utilização das janelas deslizantes significa basicamente a otimização do modelo. Houve um excelente crescimento de desempenho, seja na melhoria da predição dos dados traduzida no aumento de R2 e diminuição de MSE, bem como na redução significativa dos tempos de treinamento e teste. Este resultado sugere que, para a aplicação e para os dados utilizados neste trabalho, estes modelos projetados para lidar com dependências temporais de longo prazo não carecem de informações adicionais de tempo transcorrido, sendo mais benéfico um conjunto de dados mais enxuto, isto é, os atributos decorrentes de janelas deslizantes acabam trazendo uma complexidade difícil de lidar, o que representa os tempos de treinamento demasiadamente altos verificados quando da utilização do conjunto de dados completo.

Em resumo, diferentemente da técnica de *look-back*, imprescindível para os modelos anteriores, não é possível determinar, com base nos resultados apresentados, se a utilização da técnica de janelas deslizantes no contexto da previsão de uma série temporal por meio

Tabela 23 – Comparação dos resultados com e sem janela deslizante

Modelo	Parâmetro	Conjunto de Dados	
		Completo	sem janela deslizante
ANN	R2	0.981	0.977
	MSE	13.446	16.185
	Tempo de treinamento	2min 36s	2min 26s
	Tempo de previsão	802 ms	721 ms
CNN	R2	0.983	0.982
	MSE	11.774	12.844
	Tempo de treinamento	1min 22s	6.11 s
	Tempo de previsão	1.01 s	698 ms
LSTM	R2	0.983	0.997
	MSE	12.154	1.993
	Tempo de treinamento	20min 32s	3min 24s
	Tempo de previsão	4.15 s	1.58 s
GRU	R2	0.978	1.000
	MSE	15.345	0.021
	Tempo de treinamento	16min 41s	3min 46s
	Tempo de previsão	2.21 s	1.08 s

Fonte: do próprio autor.

das redes neurais é dispensável ou não. Enquanto a sua não-utilização trouxe uma melhoria significativa para modelos como as LSTM e as GRU em todos os critérios adotados, o modelo ANN aparentemente depende das janelas deslizantes e ainda assim consegue métricas de avaliação satisfatórias ainda num tempo de processamento inferior aos LSTM e GRU otimizados.

Portanto, a escolha entre a utilização ou não da técnica cai num problema de custo-benefício entre necessidade de precisão e tempo de processamento. Se a precisão absoluta não for um critério determinante na concepção do preditor, a melhor escolha entre as redes neurais seria pela não-utilização dos atributos temporais no conjunto de dados e a utilização do modelo CNN para realizar o treinamento dos dados e a sua predição.

6 CONCLUSÕES

Dentre as variáveis que compõem os requisitos para a realização do planejamento de sistemas elétricos de potência destaca-se a carga elétrica ou a demanda de consumo, que é a medida de potência ativa utilizada pelo consumidor, seja de ele de qualquer natureza (residencial, industrial, etc). Desse modo, é fundamental que, para o planejamento otimizado dos recursos dos sistemas de geração e transmissão, a previsão da carga seja a mais apurada possível, sendo capaz de indicar toda a dinâmica de variação temporal ao longo das horas do dia, dos dias dos meses e assim por diante.

Este trabalho buscou contribuir na busca de um preditor de carga por meio do estudo de alguns modelos regressores consagrados, baseados em técnicas de inteligência artificial. Foram escolhidos e testados os modelos de regressão linear e polinominal, modelos baseados em SVM, árvores de decisão (*Random Forest* e XGBoost) e redes neurais (ANN, CNN, LSTM e CNN. Os dados utilizados foram obtidos por meio da medição da carga no enrolamento primário dos transformadores 230/69/13,8 kV - 4x100 MVA da Subestação Porto Velho, na capital do estado de Rondônia.

Os dados obtidos foram submetidos a diversas técnicas de tratamento de dados, visando eliminar inconsistências que podem comprometer a qualidade dos resultados das previsões. Características como fatores climáticos e de tipo de dia (se a medição corresponde a dia útil ou não) foram incluídas ao conjunto de dados, a fim de fornecer informação adicional aos modelos.

A análise exploratória dos dados indicou padrões no conjunto de dados que subsidiou a escolha da arquitetura dos modelos preditores adotados, entre eles a correlação moderada para fraca da carga, variável-alvo dos modelos, para com os atributos climáticos e de característica de dia. A correlação entre a carga e a temperatura do ar foi de 0.276, foi a maior positiva e correlação entre a carga e a pressão atmosférica foi de -0.454, sendo a maior correlação negativa.

Por meio do Teste de Dickey-Fuller Aumentado, foi confirmada a estacionariedade da série temporal, ao obter um valor de ADF igual a -19.3 com *p-value* igual a zero.

A autocorrelação, por sua vez, foi confirmada por meio das análises gráficas dos testes de ACF e PACF, bem como por meio do Gráfico de Latência. O teste de ACF apresentou decaimento lento, característico de séries temporais que têm uma forte dependência de valores anteriores. O PACF, por sua vez, mostrou decaimento logarítmico indicando que a autocorrelação entre a observação atual e suas observações anteriores diminui de forma abrupta. Os pontos no Gráfico de Latência majoritariamente formaram uma reta com derivada positiva, indicando orte correlação positiva entre uma observação e

a sua defasagem. Analiticamente, o resultado dos testes de Durbin-Watson, apresentando valor próximo de zero, e de Ljung-Box, apresentando valor de variável estatística alto e *p-value* igual a zero, rejeitando a hipótese nula de que os dados seriam aleatórios - comprovando que a série temporal é autocorrelacionada.

Finalmente, o teste de Normalidade dos dados de Shapiro-Wilk apresentou valor estatístico de 0.996 e *p-value* igual a zero, enquanto o teste de Anderson-Darling, um valor de 84,2. Assumiu-se, portanto que os dados não se ajustam a uma distribuição dos dados.

Estes testes levaram à rejeição de modelos que partem do pressuposto da normalidade dos dados e à adoção dos modelos mais flexíveis a esta característica como os de regressão tradicional, os SVM, e aqueles baseados em *machine learning*, como as árvores de decisão e as redes neurais. Adicionalmente, a forte autocorrelação entre os dados também levou à adoção de técnicas de observabilidade temporal introjetada ao conjunto de dados, como as janelas deslizantes no caso das redes neurais e do *look-back* nos demais modelos adotados.

Inicialmente, com relação ao desempenho dos modelos com o conjunto de dados original, todos apresentaram ótimo desempenho validado por meio das métricas de avaliação, como o R2 e o MSE. Nos termos destas métricas, menor desempenho observado foi realizado pelo modelo de Regressão Polinomial, com R2 de 0,964 e MSE de 25,51. Empatados com destaque entre as melhores métricas ficaram os modelos XGBoost, CNN e LSTM, com R2 de 0.982 e MSE de 12.44, 13,49 e 12,69, respectivamente. Considerando o tempo total de treinamento e teste como critério de desempate, vence o modelo baseado em árvore de decisão XGBoost, por ser capaz de prever os valores futuros de carga em poucos segundos, enquanto seus concorrentes fizeram na casa dos minutos.

O modelo que obteve a melhor métrica de desempenho entre todos nesta etapa inicial, isto é, considerando o conjunto de dados completo - a carga medida ao longo do tempo e com todos os atributos adicionais no conjunto de dados, como aqueles que descrevem o dia, o clima e amostras do passado recente - foi o modelo baseado em árvore de decisão, o *Random Forest*. Este modelo obteve R2 de 0.984 e MSE de 11.23, tendo sido treinado em cerca de 19 segundo e sendo capaz de prever a janela temporal futura de 2 meses correspondente ao tamanho do conjunto de teste com ótima precisão, em menos de 6 segundos.

Este resultado já seria suficiente para definir a escolha do método preditor que melhor se ajusta aos dados disponíveis, para realizar o planejamento da operação no contexto da Eletrobras Eletronorte. Entretanto, dada a inconveniência de aquisição e a adição de dados climáticos oriundos de base de dados externa aos da empresa e do indicativo anterior de baixa correlação entre as variáveis climáticas e a variável-alvo 'carga', foram realizados testes suprimindo essas variáveis do conjunto de dados.

Os resultados dos testes de OLS e de *feature importance* corroboraram o indicativo da avaliação estatística de correlação, indicando que os atributos, embora significativos, são muito menos relevantes para a predição da carga que aquelas derivadas de observação temporal, podendo ser eliminadas em favor desta última. A confirmação definitiva veio por meio da aplicação deste conjunto de dados, ora sem os atributos climáticos com as variáveis temporais, ora com os atributos climáticos sem as variáveis temporais.

No primeiro teste, praticamente não se observou alterações nos resultados de métricas de avaliação. Ou melhor, quando variaram, foi para incrementalmente melhor nas métricas de avaliação e para satisfatoriamente melhor no tempo total de processamento. Analise gráfica apresentou difícil percepção visual de melhoria, o que indica que atributos climáticos influenciam realmente muito pouco ou quase nada.

O segundo teste, indicou que o conjunto de dados com apenas as variáveis-alvo, as climáticas e de características diárias não são capazes de prever os dados futuros com precisão. Parâmetros de métrica de avaliação caíram drasticamente em todos os modelos testados (os modelos baseados em rede neurais originalmente não levaram em conta os atributos climáticos e não utilizam a técnica específica de *look-back*).

Concluí-se assim que todos os modelos, exceto aqueles baseados em redes neurais, dependem intrinsecamente das variáveis de observação temporal para a realização da predição de carga de forma satisfatoriamente precisa.

Finalmente, um último teste foi realizado, submetendo o conjunto de teste das redes neurais a realizar a previsão somente com o *input* da variável-alvo em suas entradas, descartando as variáveis de observação temporal de amostras anteriores, as janelas deslizantes.

Os resultados apresentaram uma singela piora para o modelo de ANN, com queda de R2 de 0.981 para 0.977 e tempo de processamento mantido; melhora do desempenho computacional do modelo CNN, com R2 praticamente mantido, de 0.983 para 0.982 e tempo de treinamento caindo de 1 minuto e 22 segundos para 6 segundos.

O destaque do teste é a significativa melhora dos modelos LSTM e GRU, tanto nas métricas de avaliação quanto no tempo de processamento. O LSTM saiu de um R2 e MSE de 0.983 e 12.154, respectivamente, num tempo de treinamento superior a 20 minutos para um novo R2 de 0.997, novo MSE de 1.993 e tempo de treinamento caindo para 3 minutos e 24 segundos. Desempenho ainda melhor para o GRU, cujos parâmetros de R2, MSE e tempo de treinamento sairam de 0.978, 15.345 e 16 minutos e 41 segundos para 1, 0.021 e 3 minutos e 46 segundos, respectivamente.

Ou seja, para os modelos LSTM e GRU, a retirada das variáveis temporais significou a otimização do modelo. Reitera-se que este resultado foi obtido sendo mantida a mesma arquitetura da rede neural da execução inicial, com o conjunto completo, havendo margem

para melhoria dos tempos de treinamento em teste por meio da melhoria da arquitetura adotada e na indicação preliminar das curvas de perda destes modelos, que aparentemente é possível obter a mesma generalização com um número de épocas muito menor que a utilizada.

Dada a qualidade de precisão obtida neste último teste, a escolha do melhor preditor passou a ser um dilema entre a exatidão extrema de predição dos modelos LSTM e GRU ou a rapidez de com boa precisão do modelo *Random Forest*.

Finalmente, com base rigorosamente nas estruturas estudadas até aqui e balizado pelo resultado satisfatório obtido nas análises gráficas de predição dos valores futuros de carga e no baixo custo computacional para obter estes resultados, o modelo escolhido que que responde a pergunta norteadora deste trabalho é o *Random Forest*.

Trabalho futuro, entretanto, será aperfeiçoar as arquiteturas dos modelos baseados em redes neurais, a fim de fazê-los competir em tempo de processamento com o *Random Forest*.

Adicionalmente, após consolidada a determinação do modelo definitivo de preditor de carga, testá-lo em outros diferentes dados de carga de medição, obtendo dados de carregamento de diferentes subestações da Eletronorte.

Por fim, implementar o modelo preditor em ferramenta web e agregá-lo nos sistemas internos de gestão da supervisão ou do Sistema de Informações da Operação, sendo ferramenta decisiva na pré-operação dos sistemas da Eletrobras Eletronorte.

REFERÊNCIAS

ALTRAN, A. B. Sistema inteligente para previsão de carga multinodal em sistemas elétricos de potência. Universidade Estadual Paulista (Unesp), 2010.

ANEEL. **Procedimentos de Distribuição de Energia Elétrica no Sistema Elétrico Nacional - PRODIST**. 2016. Módulo 2 - Planejamento da Expansão do Sistema de Distribuição.

ANEEL. **Regras dos Serviços de Transmissão de Energia Elétrica**. 2022. Módulo 1 - Glossário. Rev. 3.

ASKARI, M.; KEYNIA, F. Mid-term electricity load forecasting by a new composite method based on optimal learning mlp algorithm. **IET Generation, Transmission & Distribution**, Wiley Online Library, v. 14, n. 5, p. 845–852, 2020.

BAEK, S.-M. Mid-term load pattern forecasting with recurrent artificial neural network. **Ieee Access**, IEEE, v. 7, p. 172830–172838, 2019.

BARROS, T. M. B. **Previsão de Carga-Comparação de técnicas**. 2014. Dissertação (Mestrado) — Universidade do Porto, 2014.

BORDIGNON, S. **Metodologia para previsão de carga de curtíssimo prazo considerando variáveis climáticas e auxiliando na programação de despacho de pequenas centrais hidrelétricas**. 2012. Dissertação (Mestrado), 2012.

BOX, G. E. *et al.* **Time series analysis: forecasting and control**. [*S.l.: s.n.*]: John Wiley & Sons, 2015.

CEPEL. **Sistema Aberto de Gerenciamento de Energia, Visão Geral**. 2023. Available at: <https://sage.cepel.br/index.php/pt/sage/visao-geral>.

CEPERIC, E.; CEPERIC, V.; BARIC, A. A strategy for short-term load forecasting by support vector regression machines. **IEEE Transactions on Power Systems**, IEEE, v. 28, n. 4, p. 4356–4364, 2013.

CHEN, B.-J.; CHANG, M.-W. *et al.* Load forecasting using support vector machines: A study on eunite competition 2001. **IEEE transactions on power systems**, Ieee, v. 19, n. 4, p. 1821–1830, 2004.

CHIEN, C. Fuzzy logic in control systems: fuzzy logic controller. **IEEE Trans Syst Man Cybern Part II**, v. 20, p. 429–434, 1990.

CHOW, M.-y.; TRAM, H. Application of fuzzy logic technology for spatial load forecasting. *In*: IEEE. **Proceedings of 1996 Transmission and Distribution Conference and Exposition**. [*S.l.: s.n.*], 1996. p. 608–614.

CORTES, C.; VAPNIK, V. Support-vector networks. **Machine learning**, Springer, v. 20, p. 273–297, 1995.

COSTA, C. I. d. A. **Aplicação de técnicas de Big Data à Previsão da Carga Elétrica.** 2017. Tese (Doutorado) — Universidade Federal de Itajubá, 2017.

CUNHA, L. A. d. Predição de carga de energia elétrica no brasil utilizando técnicas de deep learning. Universidade Federal de São Carlos, 2022.

DOMINGOS, S. d. O. *et al.* Previsão de carga baseada em ensemble de modelos inteligentes. 2019.

DRUCKER, H. *et al.* Support vector regression machines. **Advances in neural information processing systems**, v. 9, 1996.

DUDEK, G.; PEŁKA, P. Pattern similarity-based machine learning methods for mid-term load forecasting: A comparative study. **Applied Soft Computing**, Elsevier, v. 104, p. 107223, 2021.

FERRAZ, C. S. **Estudo de modelo de predição e correlação de indicadores para a avicultura de postura baseado em valores históricos**. 2022. Universidade de São Paulo - Instituto de Ciências Matemáticas e de Computação.

GUO, W. *et al.* Machine-learning based methods in short-term load forecasting. **The Electricity Journal**, Elsevier, v. 34, n. 1, p. 106884, 2021.

HAYKIN, S. **Redes neurais: princípios e prática**. [*S.l.: s.n.*]: Bookman Editora, 2001.

HAYKIN, S. **Neural Networks and Learning Machines - Third Edition**. [*S.l.: s.n.*]: Pearson Education, 2009.

HONG, T.; WANG, P. Artificial intelligence for load forecasting: History, illusions, and opportunities. **IEEE Power and Energy Magazine**, IEEE, v. 20, n. 3, p. 14–23, 2022.

JARDINI, J. A. *et al.* Daily load profiles for residential, commercial and industrial low voltage consumers. **IEEE Transactions on power delivery**, IEEE, v. 15, n. 1, p. 375–380, 2000.

KAGAN, N.; OLIVEIRA, C. C. B. D.; ROBBA, E. J. **Introdução aos sistemas de distribuição de energia elétrica**. [*S.l.: s.n.*]: Editora Blucher, 2021.

KERAS. **Models API**. 2023. Available at: <https://keras.io/api/models/>.

KONG, W. *et al.* Short-term residential load forecasting based on lstm recurrent neural network. **IEEE transactions on smart grid**, IEEE, v. 10, n. 1, p. 841–851, 2017.

LIN, J. *et al.* Short-term load forecasting based on lstm networks considering attention mechanism. **International Journal of Electrical Power & Energy Systems**, Elsevier, v. 137, p. 107818, 2022.

LORENA, A. C.; CARVALHO, A. C. D. Uma introdução às support vector machines. **Revista de Informática Teórica e Aplicada**, v. 14, n. 2, p. 43–67, 2007.

LU, C.-N.; WU, H.-T.; VEMURI, S. Neural network based short term load forecasting. **IEEE Transactions on Power Systems**, IEEE, v. 8, n. 1, p. 336–342, 1993.

MAMDANI, E. H.; ASSILIAN, S. An experiment in linguistic synthesis with a fuzzy logic controller. **International journal of man-machine studies**, Elsevier, v. 7, n. 1, p. 1–13, 1975.

MANSUR, M. B. Desenvolvimento de modelos de redes neurais recorrentes para previsão de demanda de energia elétrica. 2023.

NALCACI, G.; ÖZMEN, A.; WEBER, G. W. Long-term load forecasting: models based on mars, ann and lr methods. **Central European Journal of Operations Research**, Springer, v. 27, p. 1033–1049, 2019.

OLAH, C. **Understanding LSTM Networks**. 2015. Available at: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.

ONS. **Procedimentos de Rede - Submódulo 1.2**. 2020. Glossário dos Procedimentos de Rede.

ONS. **Procedimentos de Rede - Submódulo 3.9**. 2020. Validação de dados e modelos de componentes para estudos elétricos.

ONS. **O que é ONS**. 2021. Available at: <https://www.ons.org.br/paginas/sobre-o-ons/o-que-e-ons>.

ONS. **Diretrizes para Operação Elétrica com Horizonte Quadrimestral Janeiro-Abril 2023**. 2022. RT-ONS DPL/2022.

ONS. **Procedimentos de Rede - Submódulo 3.5**. 2022. Consolidação da previsão de carga para planejamento da operação eletroenergética.

ONS. **Dados Abertos - Curva de Carga Horária**. 2023. Available at: <https://dados.ons.org.br/dataset/curva-carga>.

PANDIAN, S. C. *et al.* Fuzzy approach for short term load forecasting. **Electric power systems research**, Elsevier, v. 76, n. 6-7, p. 541–548, 2006.

PAPALEXOPOULOS, A. D.; HESTERBERG, T. C. A regression-based approach to short-term system load forecasting. **IEEE Transactions on power systems**, IEEE, v. 5, n. 4, p. 1535–1547, 1990.

PARK, D. C. *et al.* Electric load forecasting using an artificial neural network. **IEEE transactions on Power Systems**, IEEE, v. 6, n. 2, p. 442–449, 1991.

RAFI, S. H. *et al.* A short-term load forecasting method using integrated cnn and lstm network. **IEEE Access**, IEEE, v. 9, p. 32436–32448, 2021.

RALSTON, F. C. Predicting central station demand and output. **Journal of the AIEE**, IEEE, v. 44, n. 1, p. 38–44, 1925.

RONDôNIA, G. do Estado de. **Decreto N° 26.739, de 28 DE dezembro DE 2021. Estabelece o calendário dos feriados e pontos facultativos de 2022 do Poder Executivo estadual no âmbito do Estado de Rondônia e dá outras providências. Diário Oficial Estado de Rondônia, Edição 254**. 2021. Available at: <https://diof.ro.gov.br/data/uploads/2021/12/doe-28-12-2021.pdf>.

RONDôNIA, G. do Estado de. **Decreto N° 27.720. Estabelece o calendário dos feriados e pontos facultativos de 2023 do Poder Executivo Estadual e dá outras providências. Diário Oficial Estado de Rondônia, Edição 245**. 2022. Available at: <https://rondonia.ro.gov.br/wp-content/uploads/2022/12/calendario-feriados.pdf>.

RONDôNIA, G. do Estado de. **ATUALIZAR**. 2023. Available at: <http://coreh.sedam.ro.gov.br/meteorologia/ATUALIZAR>.

SCIKITLEARN. **sklearn.ensemble.RandomForestRegressor**. 2023. Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html>.

SCIKITLEARN. **sklearn.preprocessing.PolynomialFeatures**. 2023. Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PolynomialFeatures.html>.

SCIKITLEARN. **sklearn.svm.SVR**. 2023. Available at: <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVR.html>.

SHARP, T. **An Introduction to Support Vector Regression (SVR)**. 2023. Available at: <https://towardsdatascience.com/an-introduction-to-support-vector-regression-svr-a3ebc1672c2>.

SHIN, T. **Understanding Feature Importance and How to Implement it in Python**. 2021. Available at: <https://towardsdatascience.com/understanding-feature-importance-and-how-to-implement-it-in-python-ff0287b20285>.

SILVA, V. R. **Previsão de carga em sistemas elétricos de potência: Um estudo com enfoque baseado no uso de métodos de *machine learning***. 2022. Universidade Estadual Paulista Julio de Mesquita Filho.

SINGH, A. K. *et al.* An overview of electricity demand forecasting techniques. **Network and complex systems**, v. 3, n. 3, p. 38–48, 2013.

SINGH, A. K. *et al.* Load forecasting techniques and methodologies: A review. *In*: IEEE. **2012 2nd International Conference on Power, Control and Embedded Systems**. [*S.l.: s.n.*], 2012. p. 1–10.

TAKAGI, T.; SUGENO, M. Fuzzy identification of systems and its applications to modeling and control. **IEEE transactions on systems, man, and cybernetics**, IEEE, n. 1, p. 116–132, 1985.

TEIXEIRA-PINTO, A.; HAREZLAK, J. **Regression and Classification Trees**. 2022. Available at: <https://bookdown.org/tpinto_home/Beyond-Additivity/regression-and-classification-trees.html>.

TENSORFLOW. **Module: tf.keras**. 2023. Available at: <https://www.tensorflow.org/api_docs/python/tf/keras>.

VAPNIK, V.; GOLOWICH, S.; SMOLA, A. Support vector method for function approximation, regression estimation and signal processing. **Advances in neural information processing systems**, v. 9, 1996.

WANG, J. Q.; DU, Y.; WANG, J. Lstm based long-term energy consumption prediction with periodicity. **Energy**, Elsevier, v. 197, p. 117197, 2020.

WANG, Y. *et al.* A hybrid ensemble method for pulsar candidate classification. **Astrophysics and Space Science**, Springer, v. 364, p. 1–13, 2019.

XGBOOST. **XGBoost documentation**. 2023. Available at: <https://xgboost.readthedocs.io/en/stable/python/python_intro.html#setting-parameters>.

ZIMMERMANN, H.-J. **Fuzzy set theory—and its applications**. [*S.l.: s.n.*]: Springer Science & Business Media, 2011.