UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Campus de Sorocaba

GABRIEL MORAES MARCIANO

USO DE ALGORITMOS DE APRENDIZADO DE MÁQUINA PARA PREDIÇÃO DE CONSUMO ENERGÉTICO

Sorocaba

2024

GABRIEL MORAES MARCIANO

USO DE ALGORITMOS DE APRENDIZADO DE MÁQUINA PARA PREDIÇÃO DE CONSUMO ENERGÉTICO

Trabalho de Conclusão de Curso de Graduação apresentado ao Instituto de Ciência e Tecnologia de Sorocaba, Universidade Estadual Paulista (UNESP), como parte dos pré-requisitos para a obtenção do grau de Bacharel em Engenharia de Controle e Automação.

Orientador: Prof. Dr. Leopoldo André Dutra Lusquino Filho.

Sorocaba

2024

M319u

Marciano, Gabriel Moraes
Uso de algoritmos de aprendizado de máquina para predição de consumo energético / Gabriel Moraes Marciano. -- Sorocaba, 2024
56 p. : il., tabs.

Trabalho de conclusão de curso (Bacharelado - Engenharia de Controle e Automação) - Universidade Estadual Paulista (UNESP), Instituto de Ciência e Tecnologia, Sorocaba
Orientador: André Dutra Lusquino Filho

1. Análise de séries temporais. 2. Consumo de energia. 3. Redes neurais (Computação). I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Biblioteca da Universidade Estadual Paulista (UNESP), Instituto de Ciência e Tecnologia, Sorocaba. Dados fornecidos pelo autor(a).

USO DE ALGORITMOS DE APRENDIZADO DE MÁQUINA PARA PREDIÇÃO DE CONSUMO ENERGÉTICO

GABRIEL MORAES MARCIANO

ESTE TRABALHO DE GRADUAÇÃO FOI JULGADO ADEQUADO COMO PARTE DO REQUISITO PARA A OBTENÇÃO DO GRAU DE BACHAREL EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

Prof. Dr. Everson Martins

Coordenador

BANCA EXAMINADORA:

Prof. Dr. LEOPOLDO ANDRÉ DUTRA LUSQUINO FILHO

Orientador / UNESP-Campus de Sorocaba

Prof. Dr. LUIS ARMANDO DE ORO ARENAS

Avaliador / UNESP-Campus de Sorocaba

Eng. RAFAEL TURELLA ARAUJO CARNEIRO

Avaliador / Externo

LUIZ FERNANDO DOS REIS OLIVEIRA

Avaliador / Externo

RESUMO

O presente trabalho propõe um estudo abrangente que investiga a eficácia do uso de redes neurais especializadas em análise de séries temporais, em particular a arquitetura LSTM (*Long Short-Term Memory*), para a predição de consumo energético em uma região específica. A predição precisa do consumo de energia é de suma importância em diversos setores, desde o planejamento de recursos até a otimização da distribuição e alocação de energia, fatores estes que tornam-se cada vez mais importantes no cenário global de mudanças climáticas. Foram realizados diversos experimentos em cenários distintos, a fim de avaliar a robustez da LSTM em comparação com métodos autorregressivos tradicionais, em relação tanto ao erro, quanto à similaridade entre as séries temporais previstas e a série temporal.A realização deste processo buscou fornecer reflexões valiosas sobre a viabilidade e eficácia do uso de redes neurais na predição de consumo energético. Os resultados obtidos demonstram que em cenários caracterizados por comportamentos não-lineares, as redes neurais possuem notável vantagem. Essas descobertas podem contribuir tanto para o avanço da pesquisa nessa área, quanto para informar decisões práticas relacionadas ao planejamento e gestão de recursos energéticos em nível regional.

Palavras-chave: séries temporais; LSTM; consumo energético; modelo preditivo; redes neurais recorrentes.

ABSTRACT

This paper proposes a comprehensive study investigating the effectiveness of using specialized neural networks in time series analysis, particularly the LSTM (Long Short-Term Memory) architecture, for predicting energy consumption in a specific region. Accurate prediction of energy consumption is of paramount importance in various sectors, from resource planning to optimizing energy distribution and allocation, factors that are becoming increasingly crucial in the global scenario of climate change. Several experiments were conducted in different scenarios to assess the robustness of LSTM compared to traditional autoregressive methods, in terms of both error and similarity between the predicted time series and the actual time series. The purpose of this process was to provide valuable insights into the feasibility and effectiveness of using neural networks in energy consumption prediction. The results demonstrate that in scenarios characterized by nonlinear behaviors, neural networks have a notable advantage. These findings can contribute both to advancing research in this area and to informing practical decisions related to regional-level energy resource planning and management.

Keywords: time series; LSTM; energy consumption; predictive model; neural networks.

SUMÁRIO

1 INTRODUÇÃO **6**
2 REVISÃO BIBLIOGRÁFICA **9**
2.1 PREDIÇÃO DE SÉRIES TEMPORAIS 9
2.2 MACHINE LEARNING APLICADO À ANÁLISE DE SISTEMAS ELÉTRICOS 11
2.2.1 ANÁLISE EXPLORATÓRIA DE DADOS 12
2.2.2 REDES NEURAIS ARTIFICIAIS 12
2.2.2.1 Neurônio *13*
2.2.2.2 Camada de um modelo *14*
2.2.2.3 Gradiente *14*
2.2.2.4 Treinamento de um modelo *14*
2.2.2.5 Perda do modelo *15*
2.2.2.6 Forward propagation *16*
2.2.2.7 Backpropagation *17*
2.2.3 REDES NEURAIS DEEP *17*
2.2.4 REDES NEURAIS RECORRENTES *18*
2.2.4.1 Rede LSTM *18*
2.2.4.2 GRU e GRU bidirecional *19*
2.2.5 ARIMA *21*
2.3 USO DE REDES NEURAIS RECORRENTES E DEEP LEARNING PARA PREDIÇÃO DE CONSUMO ENERGÉTICO 21
3 ABORDAGEM METODOLÓGICA **24**
3.1 DAS ESPECIFICAÇÕES TÉCNICAS 24
3.2 BIBLIOTECAS UTILIZADAS 24
3.3 DO DATASET UTILIZADO 25
3.4 DA AVALIAÇÃO DOS RESULTADOS OBTIDOS 29
4 RESULTADOS **31**
4.1 UMA CAMADA LSTM COM 80% DO DATASET PARA TREINO E 20% PARA TESTE 31
4.2 UMA CAMADA LSTM COM 70% DO DATASET PARA TREINO E 30% PARA TESTE 32
4.3 TRÊS CAMADAS LSTM COM 80% DO DATASET PARA TREINO E 20% PARA TESTE 33
4.5 CAMADA LSTM COM FUNÇÃO DE ATIVAÇÃO RELU E 80% DO DATASET PARA TREINO E 20% PARA TESTE 35
4.6 CAMADA LSTM COM FUNÇÃO DE ATIVAÇÃO RELU E 70% DO DATASET PARA TREINO E 30% PARA TESTE 36
4.7 GRU BIDIRECIONAL E 80% DO DATASET PARA TREINO E 20% PARA TESTE 37
4.8 GRU BIDIRECIONAL E 70% DO DATASET PARA TREINO E 30% PARA TESTE 38
4.9 GRU COM CAMADA DE ATENÇÃO E 80% DO DATASET PARA TREINO E 20% PARA TESTE 39
4.10 GRU COM CAMADA DE ATENÇÃO E 70% DO DATASET PARA TREINO E 30%

PARA TESTE 40
4.11 ARIMA (2,0,4) E 80% DO DATASET PARA TREINO E 20% PARA TESTE 41
4.12 ARIMA (2,0,4) E 70% DO DATASET PARA TREINO E 30% PARA TESTE 42
4.13 ARIMA (1,0,1) E 80% DO DATASET PARA TREINO E 20% PARA TESTE 43
4.14 ARIMA (1,0,1) E 70% DO DATASET PARA TREINO E 30% PARA TESTE 44
4.15 ARIMA (3,0,3) E 80% DO DATASET PARA TREINO E 20% PARA TESTE 45
4.16 ARIMA (3,0,3) E 70% DO DATASET PARA TREINO E 30% PARA TESTE 46
4.17 RESULTADOS FINAIS CONSOLIDADOS 47
4.17.1 DATASET SEPARADO EM 80% DE TREINAMENTO E 20% DE TESTE *47*
4.17.2 DATASET SEPARADO EM 70% DE TREINAMENTO E 30% DE TESTE *48*
5 CONCLUSÃO **50**
REFERÊNCIAS **52**
APÊNDICE A - DIFERENÇA ENTRE OS RESULTADOS OBTIDOS E VALORES REAIS PARA O MODELO COM TRÊS CAMADAS LSTM E DIVISÃO DE 80% DE TREINO E 20% DE TESTE. **55**
APÊNDICE B - DIFERENÇA ENTRE OS RESULTADOS OBTIDOS E VALORES REAIS PARA O MODELO COM UMA CAMADA LSTM E DIVISÃO DE 70% DE TREINO E 30% DE TESTE. **56**

1 INTRODUÇÃO

O cenário energético atual permite o uso das mais diversas fontes de energia, sejam elas renováveis ou não. Contudo, o subproduto da maioria delas resulta em poluentes ou até mesmo em gases que potencializam o efeito estufa, como, por exemplo, a energia nuclear ou a queima de carvão, respectivamente (WANG, 2018; YAO *et al*., 2022). Além disso, algumas fontes, apesar de não produzirem poluentes, acabam por não serem constantes, como a energia solar ou a eólica, e, no momento em que produzem menos do que é consumido em um sistema, é necessário que outra fonte de energia supra-o (LIU *et al*., 2021; ROLNICK *et al*., 2022). Por fim, vale ressaltar que em uma rede de sistema, toda energia fornecida deve ser consumida, não havendo um excedente ou mesmo falta do recurso em algum ponto.

Atualmente, as previsões apontam que a temperatura média do planeta tem subido aproximadamente 0,2 °C por década, sendo prevista que aumente 1,5 °C até 2030, quando comparada às temperaturas dos períodos pré-industriais (MATTHEW, 2022). Dentro desse cenário de intensa mudança climática, é imprescindível repensarmos o modo como estruturamos nossos sistemas sociais, tanto a fim de garantir o bem-estar geral e o funcionamento pleno da sociedade, como atingir métodos mais eficientes e sustentáveis de produção. Nesta perspectiva, o uso de modelos preditivos para melhorar a eficiência de distribuição energética, visando sempre a diminuição do uso de fontes que causam a emissão de gases do efeito estufa, é um potencial candidato para colaborar na mitigação de problemas que têm assolado a sociedade atual a níveis alarmantes. Em um contexto de mudanças climáticas extremas, é importante salientar que a própria predictabilidade dos modelos vigentes torna-se mais inconsistente, uma vez que os sistemas baseiam-se, essencialmente, em informações que não mais condizem com a materialidade do mundo.

Ao se estruturar um sistema energético, três grandes motes devem guiar nosso raciocínio:(i) é desejável que demanda energética seja igual ou próxima a quantidade de energia consumida, (ii) o fornecimento de valores constantes por parte das diversas fontes de energia utilizadas (ROLNICK *et al.*, 2022), e, por fim, (iii) que essas fontes de energia sejam as mais limpas e eficientes possíveis. As principais dificuldades encontram-se na necessidade de prever tanto a demanda quanto a produção de energia e, ao tratar de energias de fontes renováveis como solar, hidrelétrica e eólica, falamos de fontes cuja produção não é constante, uma vez que dependem de elementos naturais que naturalmente apresentam flutuações

(ROLNICK *et al*., 2022). Tomando os pontos levantados, fica evidente a necessidade de modelos preditivos que possam suprir sistemas energéticos tendo como base as três variáveis apresentadas. Na literatura atual, existem diversos estudos sobre que reforçam essa ideia, além de apontar para um fator de extrema importância, supracitado: a redução de emissão de gases que potencializam o efeito estufa (KHAN *et al*., 2020; MUSBAH *et al*., 2020; ROLNICK *et al*., 2022).

Atualmente existem diversos métodos que podem ser utilizados para previsão e melhor administração de sistemas energéticos, como ANN (redes neurais artificiais), MLP (*Multilayer perceptron*), ELM (*Extreme learning machine*), árvores de decisão, *deep learning* e modelos híbridos. Invariavelmente, cada qual possui vantagens, desvantagens e cenários onde são melhor aplicados. Além disso, é necessário que sejam comparados os resultados obtidos para que as performances de cada método possam ser avaliadas (MOSAVI *et al*., 2019). Sendo assim, é importante salientar que mesmo que existam modelos extremamente efetivos para determinados cenários, eles não necessariamente podem ser aplicados a todas as situações com sua capacidade máxima.

Tomando os pontos levantados, o objetivo é desenvolver, via *Machine Learning* (ML), especificamente LSTM (*Long short-term memory*), modelos capazes de prever, a partir do consumo e de geração de energia, maneiras de tornar um sistema de alimentação, como um bairro, autossuficiente, permitindo a troca de fontes de energia, com a maior precisão possível, sempre optando por energias renováveis, reduzindo o impacto ambiental causado, ao passo que seja fornecida a quantidade exata de energia necessária.

Considerando o cenário apresentado, o presente projeto tem como objetivo o desenvolvimento de um modelo preditivo para auxílio na gestão de sistemas de abastecimento energético, utilizando-se da metodologia mais precisa para nossa atual demanda. Para tal, prezou-se primariamente pela precisão, uma vez que nossa preocupação dá-se, essencialmente, com a crescente imprecisão dos métodos atuais, resultado direto das mudanças climáticas.

O restante deste trabalho foi estruturado da seguinte forma: o Capítulo 2 deteve-se a revisão bibliográfica dos elementos basilares do projeto, abordando a literatura científica, as bibliotecas utilizadas e as principais implicações de cada um dos elementos constituintes do modelo; o Capítulo 3 trata da abordagem metodológica do projeto, mais especificamente da sua configuração de hardware, da escolha do *dataset* e dos modelos avaliativos; o Capítulo 4 se trata da apresentação dos resultados do modelo; e o Capítulo 5 apresenta a conclusão do

trabalho, com um panorama sobre a relevância dos resultados obtidos assim como planos futuros para o projeto.

2 REVISÃO BIBLIOGRÁFICA

Nesta etapa foram explorados diversos tópicos centrais relacionados ao uso de LSTM e Deep Learning na predição de consumo energético, abrangendo conceitos fundamentais como predição de séries temporais, aplicação de Machine Learning na análise de sistemas elétricos e o funcionamento de redes neurais artificiais.

Adicionalmente, foram examinados os elementos essenciais desse universo, desde a estrutura básica de um neurônio e as camadas de um modelo até questões mais avançadas, como gradiente, treinamento de modelos, perda do modelo, forward propagation e backpropagation e, por fim a arquitetura de redes neurais deep, incluindo LSTM, GRU e GRU Bidirecional, bem como o modelo ARIMA e técnicas de Análise Exploratória de Dados para aprimorar entendimento e aplicações práticas nesse contexto energético.

2.1 PREDIÇÃO DE SÉRIES TEMPORAIS

A análise de séries temporais, que se consolidou como uma disciplina fundamental desde os anos 1970, representa uma fusão sofisticada entre estatística, matemática aplicada e ciência da computação. Este campo dinâmico e interdisciplinar desempenha um papel crucial na modelagem, interpretação e previsão de fenômenos complexos que evoluem ao longo do tempo, abrangendo uma vasta gama de setores, como economia, finanças, meteorologia, engenharia e medicina (RAFTERY, 1985).

Ao estudar conjuntos de dados sequenciais coletados em intervalos regulares, a análise de séries temporais busca identificar e modelar padrões, tendências, sazonalidades e ciclos subjacentes que podem influenciar o comportamento futuro dos dados. A sazonalidade refere-se às variações cíclicas que ocorrem em períodos fixos, enquanto a tendência descreve a direção geral em que os dados estão evoluindo ao longo do tempo, podendo ser ascendente, descendente ou estável. Uma tendência ascendente indica que os valores estão aumentando consistentemente ao longo do tempo, sugerindo um crescimento ou uma melhoria contínua. Por outro lado, uma tendência descendente indica que os valores estão diminuindo consistentemente ao longo do tempo, indicando uma queda ou uma deterioração contínua. Já uma tendência estável significa que os valores permanecem relativamente constantes ao longo do tempo, sem grandes variações ou mudanças significativas (SILVA, 2016).

Um dos modelos mais influentes e amplamente adotados nesta área é o modelo Box-Jenkins, uma abordagem metodológica para a análise e modelagem de séries temporais que se baseia na utilização de modelos ARIMA (AutoRegressive Integrated Moving Average). Esta metodologia sistemática e rigorosa utiliza uma abordagem que envolve a identificação, estimação e diagnóstico de componentes auto regressivos, de média móvel e de diferenciação. O modelo ARIMA permite desenvolver modelagens robustas e precisas para prever o comportamento de séries temporais complexas, como o consumo ou geração de energia em uma região específica, contribuindo para a tomada de decisões estratégicas informadas.

No entanto, a análise de séries temporais também enfrenta várias limitações e desafios intrínsecos, incluindo a necessidade de dados de alta qualidade e completos, a presença de ruído e outliers que podem distorcer as análises, a escolha inadequada de modelos que podem levar a previsões imprecisas, e a complexidade computacional e interpretativa de alguns métodos avançados.

Para superar essas limitações e enfrentar os desafios emergentes, têm sido desenvolvidas e aplicadas várias soluções avançadas e técnicas inovadoras na análise de séries temporais. Estas incluem métodos de suavização exponencial adaptativa, modelos de séries temporais hierárquicos e estruturados, abordagens de machine learning como redes neurais e árvores de decisão, e técnicas de aprendizado profundo como redes LSTM (Long Short-Term Memory) e CNNs (Convolutional Neural Networks).

Na área de predição de séries temporais no contexto do consumo e produção de energia, existem desafios específicos decorrentes da natureza complexa e volátil dos dados energéticos. As dificuldades incluem sazonalidade, tendências não lineares, influências externas imprevisíveis, tais como condições climáticas extremas, flutuações de demanda de energia e eventos políticos ou mudanças regulatórias. Além disso, a presença de dados incompletos ou ruidosos e a necessidade de lidar com múltiplas fontes de dados heterogêneas aumentam a complexidade da modelagem.

Dentre os principais tópicos de pesquisa atuais estão o desenvolvimento de métodos robustos para lidar com a complexidade dos dados, por exemplo por meio da decomposição desses dados, e só então os aplicar a modelos. Tal método tem mostrado bons resultados, apesar de ter um nível de complexidade maior (IFTIKHAR *et al.*,2023)

Outros métodos de aprimoramento incluem capacitar modelos, como ARIMA e SARIMA, com metodologias mais robustas (ARAÚJO, 2021) e também incorporar

contextualmente mudanças políticas e em legislações também permite que modelos assumam valores cada vez mais próximos aos reais (BARROS, 2022)

2.2 MACHINE LEARNING APLICADO À ANÁLISE DE SISTEMAS ELÉTRICOS

Os algoritmos de ML permitem a criação de modelos preditivos baseados em funções matemáticas que manipulam os dados de entrada produzindo uma resposta para determinada condição. ML possui as mais diversas aplicações, da engenharia à medicina por exemplo, e visa manipular conjuntos de dados (*datasets*) que sejam muito extensos ou que a relação entre os dados de entrada e saída seja muito complexa, como é o caso para análise de sistemas elétricos (BALOGLU *et al*., 2022).

Os sistemas de ML voltados à análise de sistemas elétricos passam por processos extremamente rigorosos, para garantir sua eficiência nas predições de consumo, geração, preço, cronogramas futuros, detecção de falhas e detecção de invasores na rede. Tem sido amplamente utilizada em energias renováveis justamente por poderem prever a demanda, mas também o quanto será produzido, uma vez que energia eólica ou solar, por exemplo, acabam não sendo constantes em sua produção, uma vez que dependem de elementos naturais. (HOSSAIN *et al*., 2019).

É possível encontrar os mais diversos usos para ML neste domínio. Por exemplo, predileções de usuários em uma rede elétrica (LI *et al*., 2011), compreender a tarifa que será cobrada de uma residência e prevendo a fonte de energia utilizada (REMANI *et al*., 2019).

2.2.1 ANÁLISE EXPLORATÓRIA DE DADOS

A Análise Exploratória de Dados, dentro da estatística, permite a sumarização das principais características de um banco de dados, muitas vezes aliados a modelos estatísticos, tendo como intuito principal uma obtenção mais precisa de dados que fuja a pré concepções estabelecidas por modelos formais (TUKEY, 1977).

O uso da Análise Exploratória de Dados é essencial no desenvolvimento de uma rede neural precisa e bem-treinada, uma vez que torna-se possível pelo método observar melhor a tendência e qualidade dos inputs informacionais fornecidos à rede, além de tornar possível a medição de sua qualidade final ao compararmos com os gráficos e modelos estatísticos já presentes sobre os temas em questão.

2.2.2 REDES NEURAIS ARTIFICIAIS

Redes neurais artificiais (RNAs) são sistemas de inteligência artificial inspirados na estrutura e funcionamento dos neurônios do cérebro humano. Elas consistem em um conjunto de unidades de processamento interconectadas, chamadas neurônios ou nós, que trabalham coletivamente para resolver tarefas complexas de aprendizado e reconhecimento de padrões. A estrutura básica de uma RNA inclui uma camada de entrada, que recebe os dados iniciais, seguida por uma ou mais camadas ocultas que realizam o processamento interno, e finalmente uma camada de saída que produz o resultado final após o processamento.

O funcionamento de uma RNA é baseado em dois processos principais: propagação direta (forward propagation) e propagação inversa (backpropagation). Durante o treinamento, a rede ajusta os pesos das conexões entre os neurônios para minimizar a diferença entre as saídas previstas e as saídas reais, permitindo que ela aprenda a partir dos dados.

As RNAs têm uma ampla gama de aplicações em diversos campos. Elas são frequentemente utilizadas para reconhecimento de padrões, como classificação de imagens e reconhecimento de voz, além de detecção de fraudes. Também são aplicadas em processamento de linguagem natural, incluindo tradução automática, geração de texto e análise de sentimentos. Além disso, são usadas em previsão e análise, como previsão de séries temporais, modelagem financeira e otimização de processos industriais.

No entanto, as RNAs também apresentam desafios. Um problema comum é o overfitting, onde a rede se torna muito complexa e memoriza os dados de treinamento, falhando em generalizar para novos dados. Além disso, em redes profundas, pode ser difícil

entender como a rede toma decisões ou explicar seu funcionamento, o que é conhecido como problema de interpretabilidade. Outros desafios incluem o requerimento de grandes volumes de dados rotulados para treinamento eficaz e o custo computacional elevado, já que treinar e manter redes profundas pode ser muito intensivo em termos de recursos computacionais.

2.2.2.1 Neurônio

Um neurônio em redes neurais artificiais, especialmente em redes neurais *deep*, é uma unidade computacional fundamental que processa e transmite informações. Inspirado no funcionamento dos neurônios biológicos encontrados no cérebro humano, um neurônio artificial é projetado para realizar operações matemáticas simples, mas poderosas, em dados de entrada para produzir uma saída. O neurônio artificial é composto por quatro componentes principais:

1. **Entradas:** Cada neurônio recebe um conjunto de entradas, que podem ser dados brutos, características extraídas ou saídas de outros neurônios.
2. **Pesos:** Cada entrada é associada a um peso, que representa a importância ou contribuição relativa daquela entrada para a saída final do neurônio. Os pesos são ajustados durante o treinamento da rede neural para melhorar o desempenho do modelo.
3. **Viés**: é um termo de viés que pode ser adicionado para ajustar o ponto de operação do neurônio
4. **Função de Ativação:** Após calcular a soma ponderada das entradas e pesos, o neurônio aplica uma função de ativação para produzir a saída. Esta função não-linear é crucial para a capacidade da rede neural de aprender e representar relações complexas nos dados.

De forma generalizada, processo de um neurônio pode ser descrito matematicamente da seguinte forma:

$$Saída \ = \ \Sigma_{i=1}^{n}\left(\left(Entrada_{i} \ * \ Peso_{i}\right) + Viés\right)$$

Os neurônios são organizados em camadas dentro de uma rede neural, e a interconexão entre essas camadas permite que a rede aprenda e represente características hierárquicas e abstratas dos dados, tornando-se capaz de realizar tarefas complexas de

aprendizado e generalização. Em redes neurais deep, milhões ou até bilhões de neurônios e suas conexões trabalham juntos para processar grandes volumes de dados e aprender representações cada vez mais complexas e úteis.

2.2.2.2 Camada de um modelo

Refere-se a um conjunto de neurônios ou unidades computacionais que realizam operações específicas sobre os dados de entrada. Em arquiteturas de redes neurais deep ou profundas, o modelo é organizado em várias camadas empilhadas sequencialmente, cada uma com um propósito e uma função específica no processamento e transformação dos dados.

2.2.2.3 Gradiente

Em redes neurais deep, o gradiente refere-se a uma medida vetorial que aponta na direção de maior crescimento de uma função. Mais especificamente, é utilizado para representar a inclinação ou a taxa de variação de uma função em relação às mudanças em seus parâmetros, como os pesos e os vieses das redes neurais.

O conceito de gradiente é fundamental no processo de treinamento de redes neurais através do algoritmo de otimização conhecido como gradiente descendente. Este algoritmo ajusta iterativamente os parâmetros do modelo (pesos e vieses) para minimizar uma função de perda, que quantifica o erro entre as previsões do modelo e os valores reais nos dados de treinamento.

O gradiente é calculado usando o cálculo diferencial e representa a derivada parcial da função de perda em relação a cada parâmetro do modelo. Em termos simples, o gradiente indica a direção e a magnitude do erro que o modelo está cometendo com base em uma pequena mudança nos parâmetros. O objetivo do gradiente descendente é ajustar os parâmetros na direção oposta ao gradiente, ou seja, na direção que minimiza a função de perda.

2.2.2.4 Treinamento de um modelo

O treinamento de um modelo em aprendizado de máquina é o processo pelo qual os parâmetros do modelo são ajustados com base nos dados de treinamento fornecidos.

Durante esse processo, o modelo "aprende" a representar e generalizar padrões a partir dos dados. Inicialmente, os parâmetros do modelo são geralmente inicializados com valores aleatórios ou pré-treinados, dependendo da arquitetura e da natureza da tarefa. Em seguida, os dados de treinamento são alimentados ao modelo em lotes ou sequências, e o modelo faz previsões com base nos parâmetros atuais. Posteriormente, é calculada a diferença entre as previsões do modelo e os valores reais usando uma função de perda, que quantifica o erro do modelo em relação aos dados de treinamento. Para ajustar e melhorar o modelo, otimizadores como o gradiente descendente são utilizados para atualizar os parâmetros do modelo de acordo com o erro calculado pela função de perda, com o objetivo de minimizá-lo. Este processo de alimentação dos dados, cálculo da perda e atualização dos parâmetros é repetido várias vezes em múltiplas iterações, conhecidas como épocas, até que a perda converge para um valor mínimo ou até que o desempenho do modelo atenda a um critério desejado.

Após o treinamento, o modelo é avaliado usando um conjunto de dados de teste ou validação separado para verificar sua capacidade de generalização e desempenho em dados não vistos. O sucesso do treinamento é influenciado por diversos fatores, como a qualidade e quantidade dos dados de treinamento, a escolha adequada da arquitetura do modelo, a função de perda, o otimizador e os hiperparâmetros. Um treinamento bem-sucedido resulta em um modelo capaz de fazer previsões precisas e úteis sobre novos dados, enquanto um treinamento inadequado pode levar a um modelo que não generaliza bem ou que apresenta super ajuste aos dados de treinamento.

2.2.2.5 Perda do modelo

Em modelos preditivos, como os baseados em aprendizado profundo, como o Long Short-Term Memory (LSTM), a perda (ou "loss", em inglês) é uma métrica fundamental para avaliar o desempenho do modelo. A perda de treinamento é calculada com base nos dados de treinamento e representa o erro médio entre as previsões do modelo e os valores reais durante o treinamento. Se essa perda diminui constantemente ao longo do tempo, geralmente indica que o modelo está aprendendo bem com os dados de treinamento. Por outro lado, a perda de teste é calculada usando um conjunto de dados separado, chamado conjunto de teste ou validação, e representa o erro médio entre as previsões do modelo e os valores reais nesse conjunto. É uma métrica crucial para avaliar o desempenho do modelo em dados não vistos, ou seja, em situações em que o modelo não foi treinado (BISHOP, 2006).

A interpretação de perda do modelo vem da perda de treinamento e perda de teste, e pode ser interpretado da seguinte forma:

- Baixa perda de treinamento e baixa perda de teste: indica que o modelo está performando bem tanto nos dados de treinamento quanto nos dados de teste, é o cenário ideal e sugere que o modelo está bem ajustado e generaliza bem para novos dados.
- Baixa perda de treinamento, alta perda de teste: indica que o modelo está super ajustado aos dados de treinamento, ou seja, o modelo está aprendendo muito bem os dados de treinamento, mas não está conseguindo generalizar bem para novos dados.
- Alta perda de treinamento e alta perda de teste: indica que o modelo não está performando bem nem nos dados de treinamento nem nos dados de teste. pode ser um sinal de que o modelo é muito simples para capturar a complexidade dos dados ou que há algum problema com os dados ou o modelo em si.

2.2.2.6 Forward propagation

A forward propagation é a fase inicial do processo de treinamento de uma rede neural deep. Nesta etapa, os dados de entrada são alimentados na rede neural, e as informações são transmitidas através das diferentes camadas da rede, passando pelos neurônios e suas funções de ativação, até chegar à camada de saída, onde são geradas as previsões ou estimativas do modelo.

Durante a propagação direta, cada neurônio em uma camada recebe as entradas dos neurônios da camada anterior, multiplica essas entradas pelos pesos associados e aplica uma função de ativação para produzir uma saída. Esta saída é então propagada como entrada para os neurônios da próxima camada, e o processo é repetido camada após camada até que as previsões finais sejam geradas na camada de saída.

A propagação direta é essencial para computar as previsões iniciais do modelo com base nos pesos e nos vieses atuais da rede neural, permitindo assim avaliar o desempenho inicial do modelo antes de iniciar o processo de *backpropagation* e ajustar os pesos durante o treinamento. Esta fase é fundamental para a capacidade da rede neural de processar informações, aprender a partir dos dados e realizar tarefas de aprendizado e inferência complexas em diversas aplicações, como classificação, regressão, reconhecimento de padrões, entre outras.

2.2.2.7 Backpropagation

É um algoritmo fundamental para o treinamento de redes neurais deep ou profundas. Ele é utilizado para calcular o gradiente da função de perda em relação aos pesos da rede neural, permitindo atualizar os pesos de forma eficiente durante o processo de treinamento através do algoritmo de otimização gradiente descendente.

O algoritmo de *backpropagation* utiliza a regra da cadeia do cálculo diferencial para calcular o gradiente da função de perda em relação aos pesos da rede neural. Esta regra permite decompor o gradiente da função de perda em uma série de derivadas parciais menores que podem ser calculadas em etapas sucessivas, começando pelas camadas finais da rede e retrocedendo em direção às camadas iniciais. Assim, o algoritmo propaga o erro da saída da rede para suas camadas internas, ajustando os pesos ao longo do caminho.

O processo é repetido iterativamente em múltiplas épocas durante o treinamento da rede neural, permitindo que o modelo ajuste seus pesos e aprenda a representar e generalizar os padrões nos dados de treinamento de forma eficiente. Este algoritmo é crucial para o sucesso do treinamento de redes neurais deep, pois permite que modelos complexos aprendam a partir de grandes volumes de dados e realizem tarefas sofisticadas de aprendizado e inferência.

Para garantir essa funcionalidade no treinamento de uma rede neural *deep*, é importante que o gradiente (o algoritmo usado para garantir a precisão do treinamento) funcione de forma a minimizar os erros entre os dados inseridos e os valores previstos.

Contudo, o aumento no número de camadas da rede *deep* pode torná-la mais propícia ao *overfitting* (quando o modelo torna-se incapaz de analisar por tornar-se muito dependente dos dados de treino) ou ao apagamento de gradiente (que, dada a extensão e frequência da entrada e saída dos dados, reduz a velocidade de aprendizado do modelo), problemas estes que modelos como o LSTM buscam solucionar.

2.2.3 REDES NEURAIS DEEP

Uma rede neural *deep* é uma arquitetura de rede neural artificial que se caracteriza pelas múltiplas camadas existentes entre o ponto de entrada de informações (*input gate*) e o ponto de saída (*output gate*) e seu funcionamento busca emular o funcionamento do cérebro humano (BENGIO, 2009).

Redes neurais *deep* são excelentes para lidar com problemas não-lineares de alta complexidade, visto que cada camada, definida como um "pedaço" do modelo que recebe as informações do "pedaço" anterior e envia para o próximo, da rede pode atribuir valores diferentes para as informações que são inseridas, permitindo que ao longo das muitas camadas a rede consiga encontrar padrões e identificar tendências.

2.2.4 *REDES NEURAIS RECORRENTES*

2.2.4.1 Rede LSTM

Uma rede *Long Short Term Memory* (LSTM) é uma rede neural recorrente (RNN, abreviação para *Recurring Neural Network*) que tem como objetivo solucionar o problema de desaparecimento de gradiente, que tende a acarretar em atrasos no treinamento de uma rede neural (HOCHREITER *et al*., 1997). Ele busca criar uma memória a curto prazo para redes neurais capaz de durar um tempo indefinido, criando portanto uma "memória a curto prazo longa".

A célula LSTM contém várias portas que regulam o fluxo de informação dentro da célula, decidindo o que deve ser lembrado ou esquecido e quando atualizar o estado da célula. As principais portas em uma célula LSTM são:

1. **Porta de Esquecimento (Forget Gate):**

a. Esta porta decide quais informações antigas no estado da célula devem ser mantidas ou descartadas.

b. É uma camada sigmoide que toma como entrada a concatenação da entrada atual x_t e da saída anterior h_{t-1} e produz um valor entre 0 e 1 para cada unidade da célula.

c. Um valor próximo de 0 indica que a informação correspondente deve ser esquecida, enquanto um valor próximo de 1 indica que a informação deve ser mantida.

2. **Porta de Entrada (Input Gate):**

a. Esta porta decide quais novas informações da entrada atual x_t devem ser adicionadas ao estado da célula.

b. Ela também é uma camada sigmoide seguida de uma camada tanh que produz dois vetores: um vetor de atualização um vetor candidato

c. O vetor de atualização determina quanto de cada novo valor candidato deve ser adicionado ao estado da célula, enquanto o vetor candidato contém os novos valores candidatos a serem adicionados ao estado da célula.

3. Porta de Saída (Output Gate):

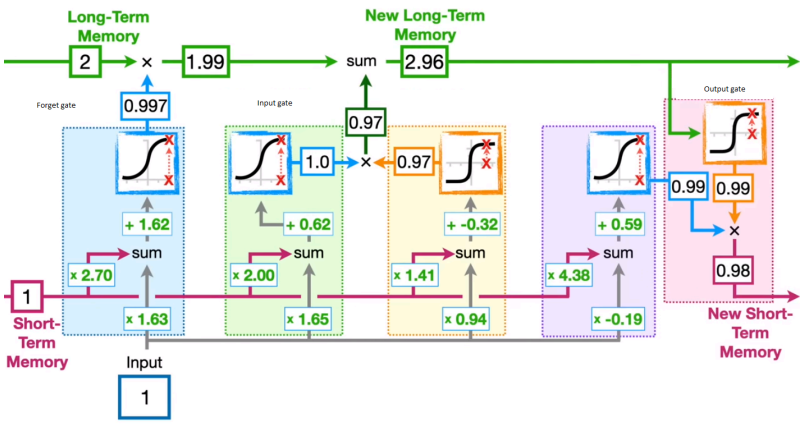
a. Esta porta decide qual parte do estado da célula deve ser exposta como saída atual h_t .

b. Ela é uma camada sigmoide que toma como entrada a concatenação da entrada atual x_t e da saída anterior h_{t-1} juntamente com o novo estado da célula e produz um valor entre 0 e 1 para cada unidade da célula.

c. Multiplica o valor resultante pelo estado da célula após aplicar a função tanh para produzir a saída atual h_t

O uso de portas em uma célula LSTM permite que o modelo aprenda a lembrar ou esquecer informações de forma seletiva ao longo do tempo, permitindo assim capturar dependências de longo prazo e realizar tarefas complexas de aprendizado em sequências de dados, como processamento de linguagem natural, reconhecimento de fala, tradução automática, entre outras aplicações.

Figura 1- Representação de uma célula de memória LSTM



Fonte: Autoria própria.

2.2.4.2 GRU e GRU bidirecional

Uma *Gated recurrent units* (GRU) é um mecanismo de controle para redes neurais similar a LSTM, utilizando-se de portões para inserir ou apagar determinadas informações (KYUNGHYUN *et al*., 2014). Por carecer de um output gate ou de um vetor de contexto para

as informações, ele resulta na obtenção de menos parâmetros do que a LSTM, apesar de apresentar um ambiente de maior controle ao treino da rede neural.

Uma estrutura de GRU Bidirecional, ou Bi-GRU, tem como objetivo atingir um melhor tratamento de dados e informações inseridos na rede neural a partir do uso de dois pontos de entrada, que circulam as informações de forma progressiva e regressiva, temporalmente falando. (DELIANIDI *et al*., 2023).

A célula GRU utiliza duas portas principais para regular o fluxo de informação dentro da célula e controlar o estado da memória: a porta de atualização (Update Gate) e a porta de reinício (Reset Gate).

1. **Porta de Atualização (Update Gate):**

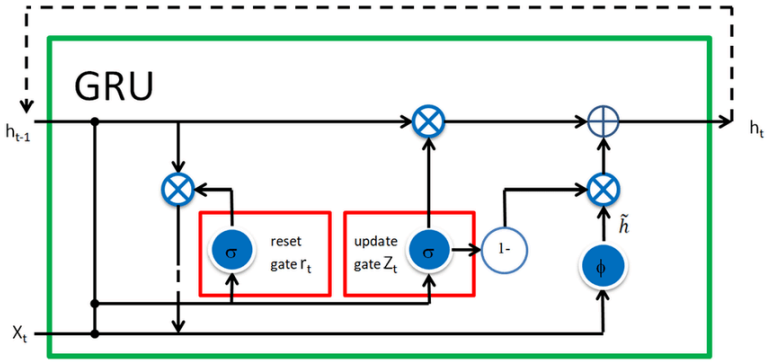
a. A porta de atualização decide quanto da informação anterior no estado da célula deve ser mantida e quanto da nova informação candidata deve ser adicionada ao estado atual.
b. É uma camada sigmóide que toma como entrada a concatenação da entrada atual e da saída anterior e produz um valor entre 0 e 1 para cada unidade da célula.
c. Um valor próximo de 1 indica que mais informações antigas devem ser mantidas, enquanto um valor próximo de 0 indica que menos informações antigas devem ser mantidas.

2. **Porta de Reinício (Reset Gate):**

a. A porta de reinício decide como a informação anterior no estado da célula deve ser combinada com a nova informação candidata.
b. É uma camada sigmoide que toma como entrada a concatenação da entrada atual e da saída anterior e produz um valor entre 0 e 1 para cada unidade da célula.
c. Um valor próximo de 1 indica que mais informações antigas devem ser consideradas, enquanto um valor próximo de 0 indica que menos informações antigas devem ser consideradas.

A célula GRU combina as informações antigas e novas de forma adaptativa e eficiente, permitindo que o modelo aprenda a representar dependências de longo prazo em sequências de dados e realizar tarefas complexas de aprendizado, como processamento de linguagem natural, reconhecimento de fala, tradução automática, entre outras aplicações. Devido à sua estrutura mais simples e eficiente, a célula GRU é frequentemente preferida em aplicações onde a velocidade de treinamento e a eficiência computacional são prioritárias.

Figura 2- Representação de uma célula de memória GRU



Fonte: Autoria própria

2.2.5 ARIMA

Os modelos ARIMA, ou *AutoRegressive Integrated Moving Average*, representam uma classe essencial de modelos estatísticos utilizados para análise e previsão de séries temporais. Esses modelos são valiosos para compreender e modelar padrões temporais complexos, abrangendo tendências, sazonalidades e flutuações irregulares nos dados ao longo do tempo. A sigla ARIMA é uma referência às suas três componentes principais: AutoRegressão (AR), que captura a relação linear entre a variável de interesse e suas observações passadas; a parte Integrada (I), que se refere à diferenciação da série temporal para torná-la estacionária, removendo tendências e sazonalidades; e a parte Moving Average (MA), que modela a relação entre o erro de previsão atual e os erros de previsão passados, abordando flutuações irregulares e aleatórias na série temporal (HYNDMAN; ATHANASOPOULOS, 2018).

Esses modelos são especificados por meio de três parâmetros principais: p, d e q, que correspondem às ordens das componentes AR, I e MA, respectivamente. Amplamente utilizados em diversos domínios, como economia, finanças, meteorologia e saúde, os modelos ARIMA fornecem uma estrutura flexível e poderosa para modelar e prever séries temporais com diferentes padrões e características. (BOX *et al*., 2015).

2.3 USO DE REDES NEURAIS RECORRENTES E DEEP LEARNING PARA PREDIÇÃO DE CONSUMO ENERGÉTICO

A previsão precisa do consumo energético desempenha um papel crucial na promoção da eficiência operacional e na otimização de recursos em setores variados, que vão desde

redes elétricas e edifícios inteligentes até sistemas industriais. O advento da tecnologia de deep learning, especificamente através da utilização de Long Short-Term Memory (LSTM), tem promovido avanços notáveis na acurácia das previsões de consumo energético (BERRIEL, 2017).

Pesquisas recentes evidenciam a eficácia das LSTMs em abordar a natureza intrinsecamente não-linear e dinâmica dos dados relacionados à produção de energia solar. Essa abordagem tem se mostrado superior em termos de precisão quando comparada aos métodos convencionais (ZHANG, 2018). Além disso, comparações diretas entre as LSTMs e outras arquiteturas de redes neurais destacam consistentemente o desempenho superior das LSTMs em previsões de consumo energético (SON, 2021).

O emprego de deep learning, em especial as LSTMs, tem revelado um potencial significativo na predição precisa do consumo energético em múltiplas aplicações. Estes estudos não apenas corroboram melhorias substanciais na precisão das previsões, mas também ressaltam a habilidade distintiva das LSTMs em identificar e modelar padrões complexos e dinâmicos nos dados de consumo energético. Contudo, desafios persistem, incluindo a escolha apropriada de arquiteturas de rede, o tratamento eficaz de dados contaminados por ruídos e a necessidade de tornar os modelos mais interpretáveis.

Consequentemente, é imperativo que pesquisas futuras se concentrem em abordar essas questões pendentes, visando aprimorar ainda mais as técnicas de deep learning e maximizar seu potencial na predição precisa do consumo energético.

A análise de séries temporais, que se consolidou como uma disciplina fundamental desde os anos 1970, representa uma fusão sofisticada entre estatística, matemática aplicada e ciência da computação. Este campo dinâmico e interdisciplinar desempenha um papel crucial na modelagem, interpretação e previsão de fenômenos complexos que evoluem ao longo do tempo, abrangendo uma vasta gama de setores, como economia, finanças, meteorologia, engenharia e medicina (RAFTERY, 1985). .

Ao estudar conjuntos de dados sequenciais coletados em intervalos regulares, a análise de séries temporais busca identificar e modelar padrões, tendências, sazonalidades e ciclos subjacentes que podem influenciar o comportamento futuro dos dados. A sazonalidade refere-se às variações cíclicas que ocorrem em períodos fixos, enquanto a tendência descreve a direção geral em que os dados estão evoluindo ao longo do tempo, podendo ser ascendente, descendente ou estável. Uma tendência ascendente indica que os valores estão aumentando consistentemente ao longo do tempo, sugerindo um crescimento ou uma melhoria contínua.

Por outro lado, uma tendência descendente indica que os valores estão diminuindo consistentemente ao longo do tempo, indicando uma queda ou uma deterioração contínua. Já uma tendência estável significa que os valores permanecem relativamente constantes ao longo do tempo, sem grandes variações ou mudanças significativas (SILVA, 2016).

Um dos modelos mais influentes e amplamente adotados nesta área é o modelo Box-Jenkins, uma abordagem metodológica para a análise e modelagem de séries temporais que se baseia na utilização de modelos ARIMA (AutoRegressive Integrated Moving Average). Esta metodologia sistemática e rigorosa utiliza uma abordagem que envolve a identificação, estimação e diagnóstico de componentes auto regressivos, de média móvel e de diferenciação. O modelo ARIMA permite desenvolver modelagens robustas e precisas para prever o comportamento de séries temporais complexas, como o consumo ou geração de energia em uma região específica, contribuindo para a tomada de decisões estratégicas informadas.

No entanto, a análise de séries temporais também enfrenta várias limitações e desafios intrínsecos, incluindo a necessidade de dados de alta qualidade e completos, a presença de ruído e outliers que podem distorcer as análises, a escolha inadequada de modelos que podem levar a previsões imprecisas, e a complexidade computacional e interpretativa de alguns métodos avançados.

Para superar essas limitações e enfrentar os desafios emergentes, têm sido desenvolvidas e aplicadas várias soluções avançadas e técnicas inovadoras na análise de séries temporais. Estas incluem métodos de suavização exponencial adaptativa, modelos de séries temporais hierárquicos e estruturados, abordagens de machine learning como redes neurais e árvores de decisão, e técnicas de aprendizado profundo como redes LSTM (Long Short-Term Memory) e CNNs (Convolutional Neural Networks).

3 ABORDAGEM METODOLÓGICA

3.1 DAS ESPECIFICAÇÕES TÉCNICAS

Para o desenvolvimento dessa pesquisa, foi utilizado um laptop modelo NP 2705, produzido pela Samsung, com processador Intel(R) Core(TM) i5-4210U CPU 1.70GHz 2.40 GHz, RAM de 8 GB com sistema operacional Windows 10 PRO, versão 20H2. Para o tratamento dos dados, foi utilizado o Google Colab[1], plataforma que permite a criação de notebooks a partir de conjuntos de dados provenientes das mais diversas fontes, além de outras ferramentas capazes de oferecer o suporte necessário para a aplicação que é desenvolvida, como as linguagens de programação Python e bibliotecas que tem foco em ML e análise de dados.

3.2 BIBLIOTECAS UTILIZADAS

3.2.1 SCIKIT-LEARN[2]

O scikit-learn, ou sklearn, é uma biblioteca gratuita para a linguagem Python com um acervo voltado para o Machine Learning. Ela contém vários algoritmos para classificação, regressão e clustering, e é pensada para incorporar bancos de dados científicos e numéricos em Python, como o SciPy e o NumPy, respectivamente. Sua versatilidade e gratuidade permitem uma maior facilidade na integração e exploração das ferramentas necessárias para a criação de redes neurais voltadas para os fins pré-estabelecidos por este projeto.

3.2.2 PANDAS[3]

O pandas é uma biblioteca gratuita para linguagem Python com foco em manipulação e análise de dados. É notável para nós seu vasto repertório em *data structures* e ferramentas para manipular séries temporais e tabelas numéricas. Tendo em vista o projeto desenvolvido, o acesso a tal biblioteca mostra-se fundamental para encontrar modelos precisos de estruturação da rede neural proposta.

3.2.3 NUMPY[4]

[1] Disponível em: https://colab.research.google.com. Acesso em: 27 fev. 2024
[2] Disponível em: https://scikit-learn.org/stable/. Acesso em: 20 fev. 2024.
[3] Disponível em: https://pandas.pydata.org. Acesso em: 20 fev. 2024.
[4] Disponível em: https://numpy.org. Acesso em: 20 fev. 2024.

O NumPy é uma biblioteca gratuita para a linguagem Python que fornece suporte para funções matemáticas, matrizes, arranjos e outros elementos importantes para a manipulação dos mesmos. Ao desenvolver uma rede neural, é importante recorrer a tais recursos para que a obtenção de dados de treinamento, predições e avaliação do projeto em si ocorra da forma mais precisa possível.

3.2.4 SEABORN[5]

O Seaborn é uma biblioteca de visualização de dados em Python construída sobre o Matplotlib. Ele oferece uma interface de alto nível para criar gráficos estatísticos atrativos e informativos. O Seaborn simplifica o processo de criação de visualizações complexas, oferecendo uma ampla gama de temas e paletas de cores incorporadas, além de funções para criar diversos tipos de gráficos estatísticos, como gráficos de dispersão, de barras, de caixas e de violino.

3.2.5 PYSPARK[6]

Apache Spark é um software open-source que age como engine para análise de dados em larga escala. O PySpark, por outro lado, utiliza-se da natureza open-source do Spark para introduzir a este uma interface em Python, o que permite a integração uma integração de linguagem no desenvolvimento do projeto e permite a análise interativa dos dados fornecidos que serão fornecidos à rede neural. Seu uso mostra-se essencial dada especificamente a natureza dos dados analisados: estatísticas de redes elétricas estadunidenses apresentam um corpo de dados amplo e de grande magnitude, o que exige que estes sejam lidos, tratados e fornecidos ao sistema neural desenvolvido a partir de ferramentas aptas para tal trabalho, como é o caso do Apache Spark.

3.3 DO *DATASET* UTILIZADO

O *dataset* "Hourly energy consumption time series data"[7] é um conjunto de dados utilizado para estudos de previsão de energia e análise de séries temporais, e contém registros de potência instantânea medidas em intervalos de hora em hora ao longo de um período de tempo específico. Foi obtido pelo "PJM Interconnection", uma organização regional de

[5] Disponível em: https://seaborn.pydata.org. Acesso em: 23 fev. 2024
[6] Disponível em: https://spark.apache.org/docs/latest/api/python/index.html . Acesso em: 20 fev. 2024.
[7] Disponível em: https://l1nq.com/N3OOB. Acesso em: 20 fev.2024

transmissão de energia elétrica nos Estados Unidos que coordena a movimentação de eletricidade em vários estados do leste e centro-oeste do país. O *dataset* inclui dados de consumo de energia elétrica em diferentes regiões atendidas pela rede PJM. Apesar de o *dataset* ser conhecido no âmbito de estudantes, não foram encontrados grandes estudos que fizessem uso do conjunto de dados para desenvolver grandes pesquisas. Contudo, vale mencionar estudos desenvolvidos por comunidades online, como Kaggle, onde foram obtidos resultados sólidos fazendo uso do *dataset* em questão e topologias como LSTM[8], GRU[9]

Esses conjuntos de dados são valiosos para análises de previsão de carga, modelagem de demanda de energia, desenvolvimento de estratégias de operação e planejamento de recursos. Eles são frequentemente utilizados em pesquisas acadêmicas, análises de mercado de energia e desenvolvimento de modelos de previsão para ajudar as empresas e operadores de rede a tomar decisões informadas sobre o gerenciamento da oferta e demanda de energia elétrica.

O estudo foi desenvolvido fazendo uso da potência instantânea em MW devido à disponibilidade dos dados e à possibilidade de aplicar técnicas de aprendizado de máquina para prever variações na demanda de energia em tempo real. Embora os dados estejam em MW, é possível desenvolver modelos preditivos significativos que capturem padrões e tendências na potência instantânea, auxiliando na identificação de comportamentos dinâmicos e na otimização da operação de sistemas elétricos.

As previsões de potência instantânea têm relevância prática na gestão e operação de sistemas de energia, permitindo um planejamento mais eficiente e uma resposta mais ágil às variações na demanda. Ao entender e prever a potência necessária em diferentes momentos, é possível tomar decisões informadas sobre a geração, transmissão e distribuição de energia, contribuindo para a estabilidade e a eficiência do sistema elétrico.

Embora os dados utilizados estejam em MW, a potência instantânea oferece insights valiosos sobre o consumo energético em tempo real, possibilitando o desenvolvimento de modelos preditivos robustos e a aplicação prática das previsões no setor energético.

O *dataset* completo é composto por medição de energia oferecida pela empresa, em megaWatts, coletado hora a hora de doze subsidiárias da empresa PJM, cada uma tendo em

---

[8] Disponível em: https://www.kaggle.com/code/msripooja/hourly-energy-consumption-time-series-rnn-lstm. Acesso em: 30 fev. 2024.
[9] Disponível em: https://www.kaggle.com/code/gabrielloye/gru-vs-lstm-prediction. Acesso em: 30 fev. 2024.

vista uma determinada região, sendo elas: AEP, COMED, DAYTON, DEOK, DOM, DUQ, EKPC, FE, NI, PJM, PJME, PJMW. Para que pudesse ser desconsiderado o fator geográfico no consumo energético, foi necessário selecionar apenas uma empresa, sendo aquela que atendesse melhor ao objetivo do estudo. Para tal, é necessário que a empresa escolhida dentro do *dataset* apresente os seguintes pontos: contenha o maior número possível de registros, maior período de tempo, maior período de continuidade. Para que os pontos sejam atendidos, foram extraídas as datas máximas e mínimas dentro dos *datasets*, obtendo assim o período de medição. Aliado a isso, foi realizada uma contagem de registros. Dessa forma, sabendo o tempo total de coleta e a periodicidade com que os dados foram extraídos, é possível determinar se há continuidade dos dados obtidos. A Tabela 1 foi construída a partir desses dados.

Tabela 1 - Período de e número de medições para cada subsidiária presente no *dataset*.

Empresa	Data primeira medição	Data última medição	Período medido	Número de medições
AEP	01/10/2004	03/08/2018	13 anos 10 meses 2 dias	121.273
COMED	01/01/2011	03/08/2018	7 anos 7 meses 2 dias	66.497
DAYTON	01/10/2004	03/08/2018	13 anos 10 meses 2 dias	121.275
DEOK	01/01/2012	03/08/2018	6 anos 7 meses 2 dias	57.739
DOM	01/05/2005	03/08/2018	13 anos 3 meses 2 dias	116.189
DUQ	01/01/2005	03/08/2018	13 anos 7 meses 2 dias	119.068
EKPC	01/06/2013	03/08/2018	5 anos 2 meses 2 dias	45.334
FE	01/06/2011	03/08/2018	7 anos 2 meses 2 dias	62.874
NI	01/05/2004	01/01/2011	6 anos 8 meses	58.450
PJM	01/04/1998	01/01/2002	3 anos 9 meses	32.896
PJME	01/01/2002	03/08/2018	16 anos 7 meses 2 dias	145.366
PJMW	01/04/2002	03/08/2018	16 anos 4 meses 2 dias	143.206

Fonte: Autoria própria

A análise da Tabela 1 deixa evidente que os maiores *datasets* são das empresas AEP, DAYTON, DOM, DUQ, PJME e PJMW.

Para melhor entendimento dos dados utilizados, também foi obtido um boxplot para cada uma das empresas, de forma a compreender a distribuição dos dados utilizados. Contudo, após sua confecção foi possível determinar que apenas a análise bruta dos dados em si não permite garantir que os outliers presentes são provenientes de erros, medições que

ocorreram de forma errada ou se são de fato os valores medidos, representando picos ou baixas de consumo. Devido a esse fator, a presença de outliers foi desconsiderada para escolha do dataset.

Portanto, após as análises levantadas, foi selecionada a empresa PJME para o estudo, salientando que a série temporal escolhida é univariada, isto é, um tipo específico de série temporal que consiste em observações de uma única variável ao longo do tempo. Em outras palavras, é uma sequência de dados temporais em que apenas uma única medida é registrada em cada intervalo de tempo. Selecionado o *dataset*, foram determinadas métricas para avaliar os dados que serão utilizados, sendo eles valores máximo, mínimo e faixa das medições, média, mediana, desvio padrão, assimetria e curtose. Todas essas métricas foram utilizadas para determinar a distribuição dos dados a serem compreendidos, para que assim recebam os melhores tratamentos de forma que o método aplicado no final seja o mais adequado.

TABELA 2 - Métricas básicos para a subsidiária PJME

Mínimo	Máximo	Faixa	Média	Mediana	Desvio Padrão	Assimetria	Curtose	Nº de dados
14.544,0	62.009,0	47.465,0	32.080,2	31.421,0	6.463,99	0,7390	0,7368	145.366

Fonte: Autoria própria

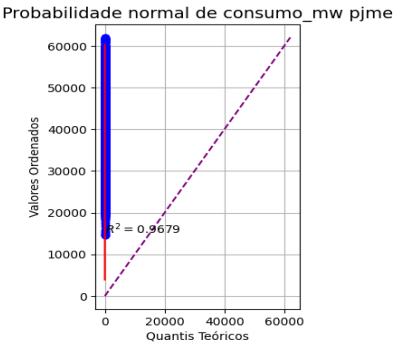
Obtidas as métricas, em seguida foi necessário analisar a distribuição dos dados para que fosse possível entender melhor sua forma, centralidade e dispersão. Para tal, foi utilizada a função distplot da biblioteca Seaborn e a análise do resultado da função evidencia o que está sendo descrito na Tabela 2 pois a média e mediana do *dataset* está onde há maior concentração de valores, por volta dos 30.000 MW.

A última característica a ser investigada no *dataset* é sua probabilidade normal. Essa etapa é fundamental para garantir a validade e interpretação correta dos resultados estatísticos e dos modelos construídos a partir desses dados. Embora a normalidade nem sempre seja necessária, sua verificação é uma etapa importante na análise exploratória de dados e na modelagem estatística. Através da visualização gráfica, obteve-se a Figura 3.

Embora a distribuição seja normal e centrada em 0, pode haver uma assimetria nos dados que não é refletida na média. Isso significa que, apesar de os dados terem uma média de 0, eles podem estar desviados para um dos lados da distribuição, o que afeta a simetria dos dados, que é possível observar no distplot do *dataset*. Além disso, um coeficiente de determinação alto (R^2 = 0,9679) sugere que a variabilidade do *dataset* é bem explicada pelo

modelo ou pela distribuição normal. Em outras palavras, cerca de 96,79% da variabilidade dos dados pode ser explicada pelo modelo de distribuição normal centrada em 0.

Figura 3 - Representação gráfica da probabilidade normal do *dataset* utilizado



Fonte: Autoria própria.

3.4 DA AVALIAÇÃO DOS RESULTADOS OBTIDOS

Para o desenvolvimento da pesquisa, foram construídos diversos modelos com o objetivo de encontrar aquele que melhor atende à situação. Fazendo uso de LSTM foram desenvolvidos três modelos distintos, cada um operando com um conjunto de teste equivalente a 80% do *dataset* original, de forma a prever os 20% restantes. Além disso, esses mesmos modelos foram foram reaplicados para 70% do *dataset* original prevendo os 30% restantes, de forma a comparar os resultados. Essas divisões foram realizadas pois assim seria possível testar a resistência do modelo em diversos cenários. Assim sendo, na divisão de 70% e 30% desejava-se determinar se o modelo apresentaria *underfitting*, de forma a determinar uma quantidade mínima de dados a serem utilizados. A distribuição de 80% e 20%, por sua vez, busca identificar se o modelo apresentaria *overfitting*.

Por fim, de forma a avaliar se a arquitetura escolhida ofereceu de fato a melhor solução, foram desenvolvidos outros três modelos que fazem uso de outras arquiteturas. Um deles consiste em um modelo baseado em GRU bidirecional, o outro em GRU com uma camada de atenção, ambos operando com a mesma divisão mencionada anteriormente. O último modelo desenvolvido é baseado na metodologia ARIMA, onde foram testados três conjuntos de valores para a ordem do modelo, o grau de diferenciação e ordem do modelo de média móvel.

Uma vez que os modelos foram executados, cada um com sua particularidade, foram obtidos os resultados e métricas que permitissem sua avaliação tanto para o conjunto de teste quanto para o conjunto de treino, sendo elas:

- erro médio absoluto (MAE): O MAE é uma métrica que calcula a média das diferenças absolutas entre os valores observados e os valores previstos. Ele fornece uma medida da precisão das previsões, sendo essencial para avaliar a acurácia dos modelos de previsão;
- erro médio quadrático: O RMSE calcula a raiz quadrada da média dos quadrados das diferenças entre os valores observados e os valores previstos. Ele é sensível a grandes erros e é útil para penalizar erros maiores, fornecendo uma medida mais robusta da precisão do modelo;
- DTW: Dynamic Time Warping é uma técnica de alinhamento de séries temporais que permite comparar séries temporais de comprimentos diferentes, levando em consideração as variações de tempo. É especialmente útil quando as séries temporais têm diferentes velocidades ou distorções temporais;
- SMAPE: Symmetric Mean Absolute Percentage Error é uma métrica que calcula a média das porcentagens absolutas de erro entre os valores observados e os valores previstos. Ele fornece uma medida da precisão das previsões, sendo simétrico em relação aos valores observados e previstos.

Além das métricas descritas, os resultados também foram representados graficamente, comparando-os ponto a ponto com os valores reais, ou seja, as menores parcelas do *dataset* original

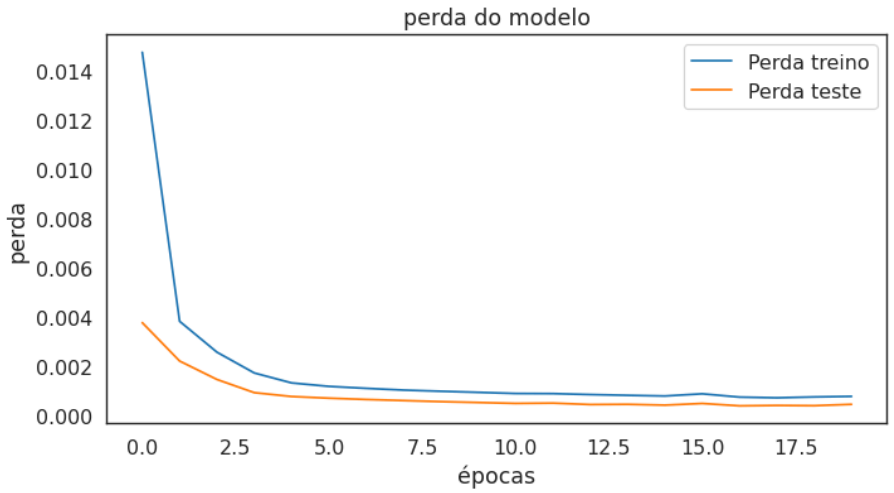
4 RESULTADOS

4.1 UMA CAMADA LSTM COM 80% DO *DATASET* PARA TREINO E 20% PARA TESTE

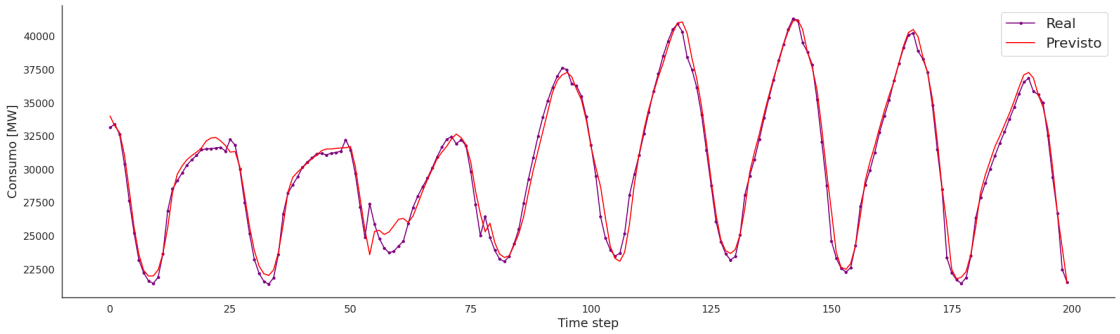
Para o modelo que faz uso de apenas uma camada LSTM, foram obtidas as seguintes representações gráficas:

Figura 4 - Representação gráfica dos resultados obtidos para o modelo com uma camada LSTM com 80% do dataset para treino e 20% para teste

a) Perda do modelo.



b) Comparação dos resultados obtidos com os reais



Fonte: Autoria própria

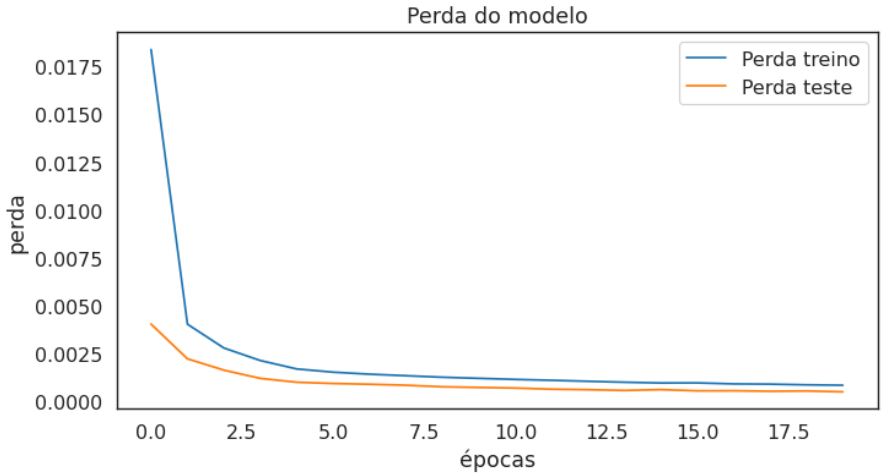
Ao comparar os dados que foram previstos com os dados que foram obtidos, é possível notar que o modelo LSTM possui uma capacidade razoável de capturar os padrões e a variabilidade nos dados de consumo de energia elétrica da rede PJME. No entanto, considerações adicionais podem ser necessárias para otimizar ainda mais o desempenho do modelo e explorar seu potencial em cenários de previsão de consumo de energia elétrica,

sendo notável salientar que o modelo é incapaz de extrapolar em condições que fogem da situação comum, como fica evidente entre os timesteps 50 e 75.

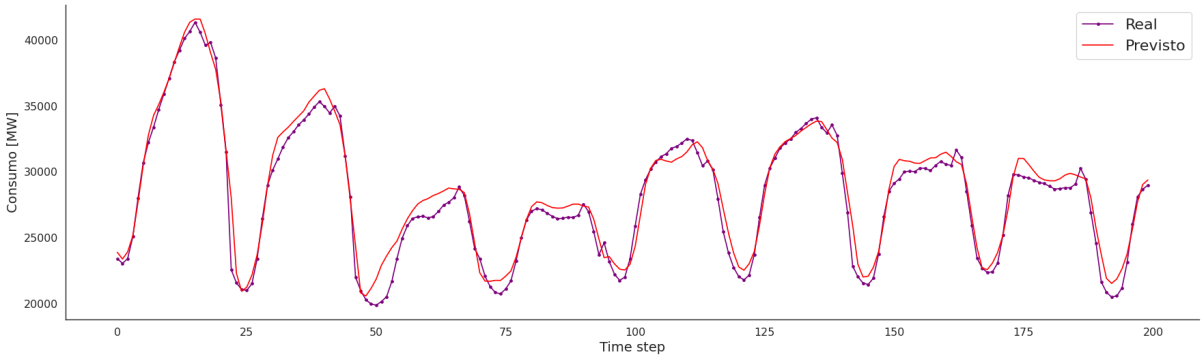
4.2 UMA CAMADA LSTM COM 70% DO *DATASET* PARA TREINO E 30% PARA TESTE

Figura 5 - Representação gráfica da perda do modelo com uma camada LSTM usando 70% do *dataset* para treino e 30% para teste.

a) Perda do modelo



b) Comparação dos resultados obtidos com os reais



Fonte: Autoria própria

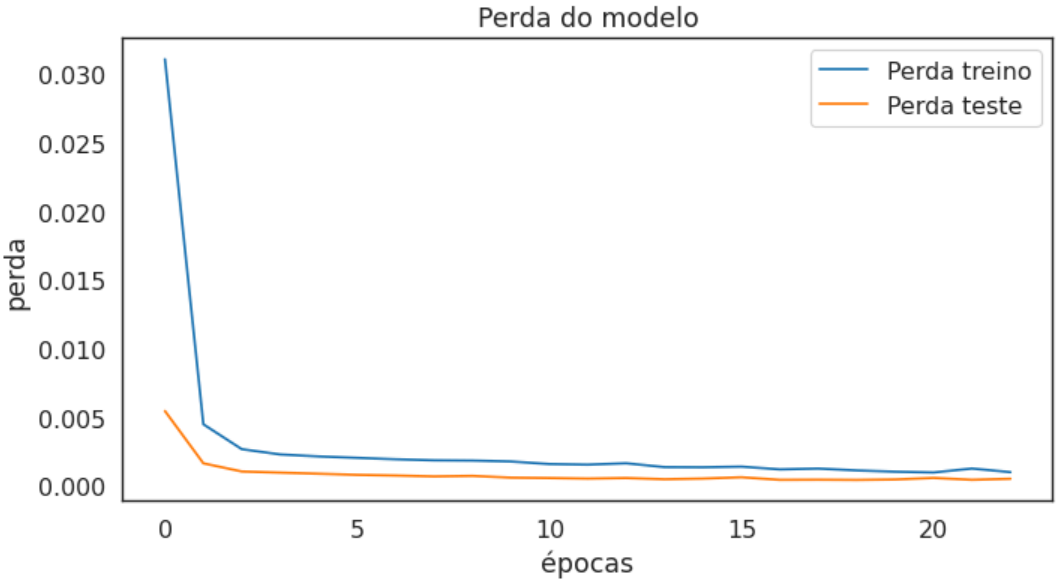
Usando 70% do *dataset* original, as previsões sofrem uma suavização, tornando-se mais genéricas. Ao mesmo tempo, a perda de precisão torna esse modelo menos eficiente.

Uma breve comparação entre os resultados dos modelos que fazem uso de apenas uma camada LSTM deixa evidente que o que faz uso de 80% do *dataset* para treinar possui maior precisão, ao passo que o modelo que faz uso de 70% permite maiores generalizações.

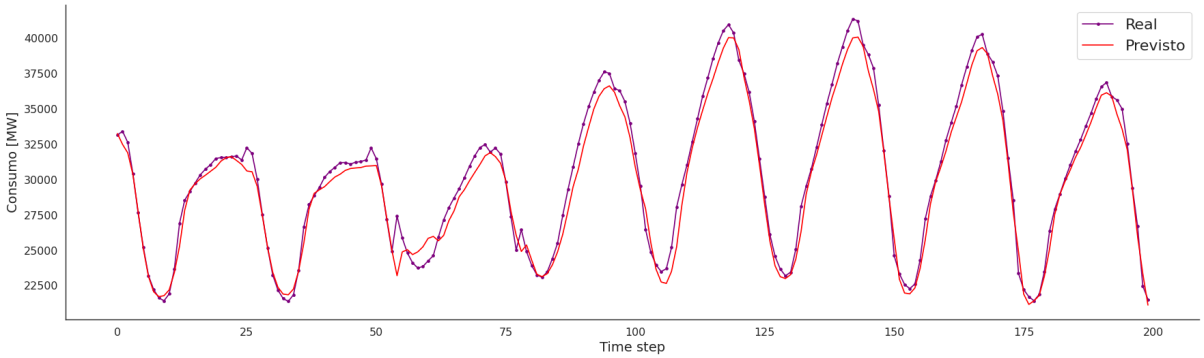
4.3 TRÊS CAMADAS LSTM COM 80% DO *DATASET* PARA TREINO E 20% PARA TESTE

Figura 6 - Perda do modelo com três camadas LSTM usando 80% do *dataset* para treino e 20% para teste.

a) Perda do modelo.



b) Comparação dos resultados obtidos com os reais



Fonte: Autoria própria

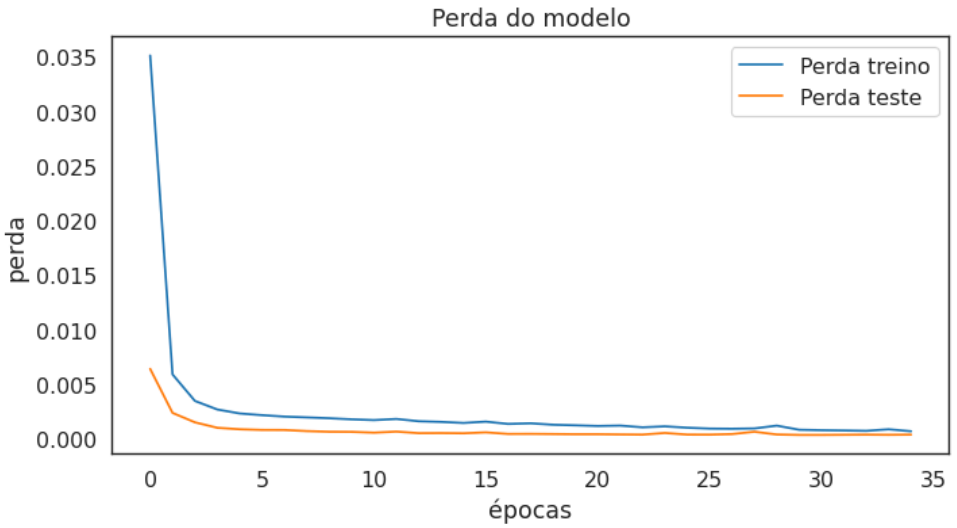
Ao adicionar mais camadas LSTM dentro de um mesmo modelo, são oferecidas mais oportunidades para o modelo aprender padrões nos dados, o que pode ajudar a evitar que este se ajuste demais aos dados de treinamento. Com isso, o modelo consegue atingir maiores similaridades com o *dataset* original, até mesmo respondendo rapidamente para anomalias, como picos fora de épocas. O uso de três camadas LSTM em um modelo pode oferecer benefícios significativos na representação de relações temporais complexas e na modelagem de sequências de dados.

Além disso, é possível notar que a perda dos conjuntos de treino e teste convergem para um mesmo valor, que, por sua vez, tende a zero.

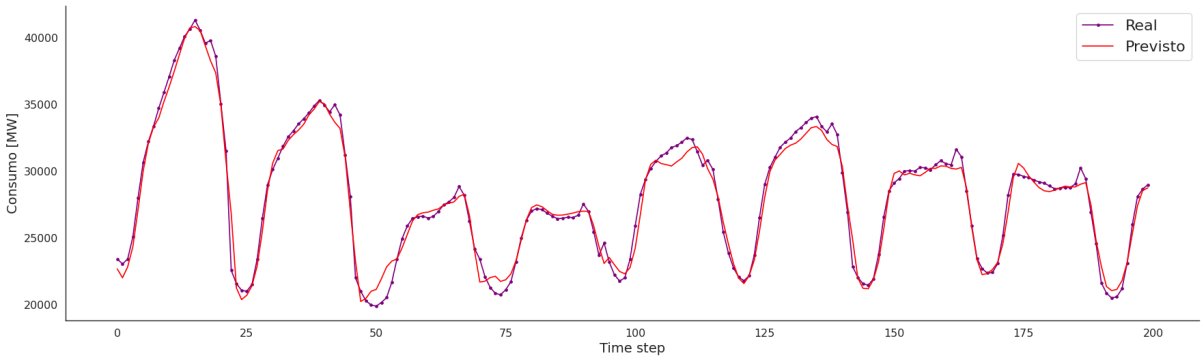
4.4 TRÊS CAMADAS LSTM COM 70% DO *DATASET* PARA TREINO E 30% PARA TESTE

Figura 7 - Perda do modelo com três camadas LSTM usando 70% do *dataset* para treino e 30% para teste.

a) Perda do modelo.



b) Comparação dos resultados obtidos com os reais



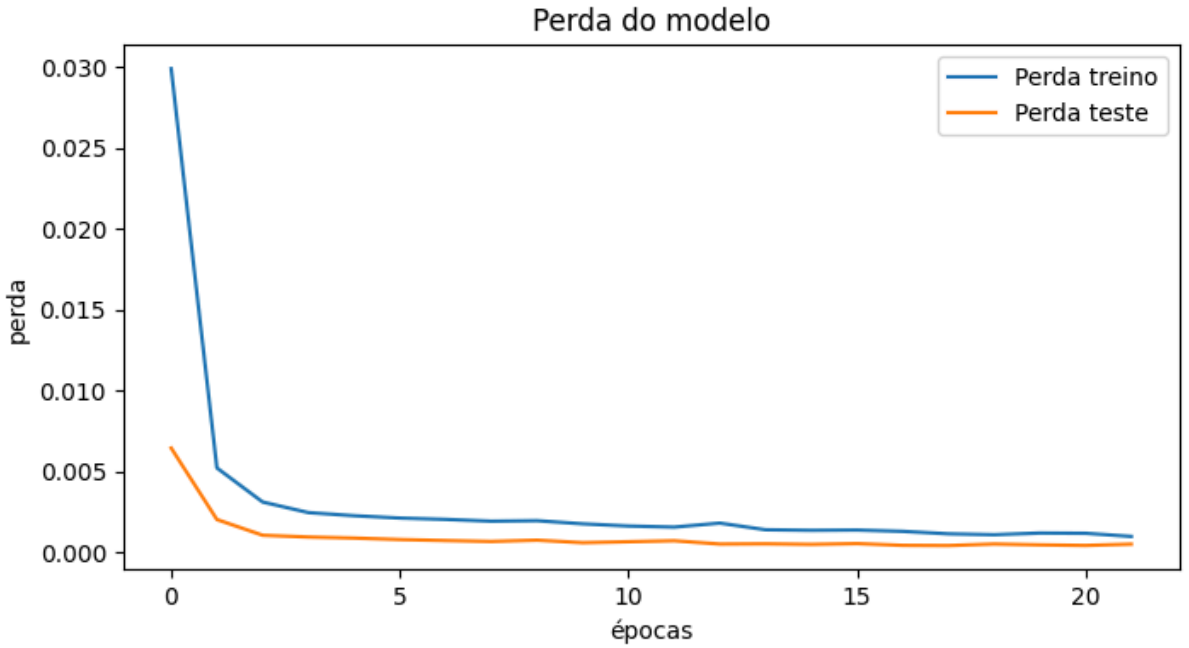
Fonte: Autoria própria

Conforme observado nos dois primeiros modelos, ao reduzir o tamanho do conjunto de dados de treinamento, os resultados obtidos para o modelo não são tão precisos quanto na versão anterior, que utiliza 80% dos dados. No entanto, o modelo resultante é mais genérico, o que implica em uma menor probabilidade de overfitting. Em outras palavras, embora a precisão possa diminuir com um conjunto de dados menor, o modelo tende a generalizar melhor para novos dados, evitando ajustes excessivos aos dados de treinamento.

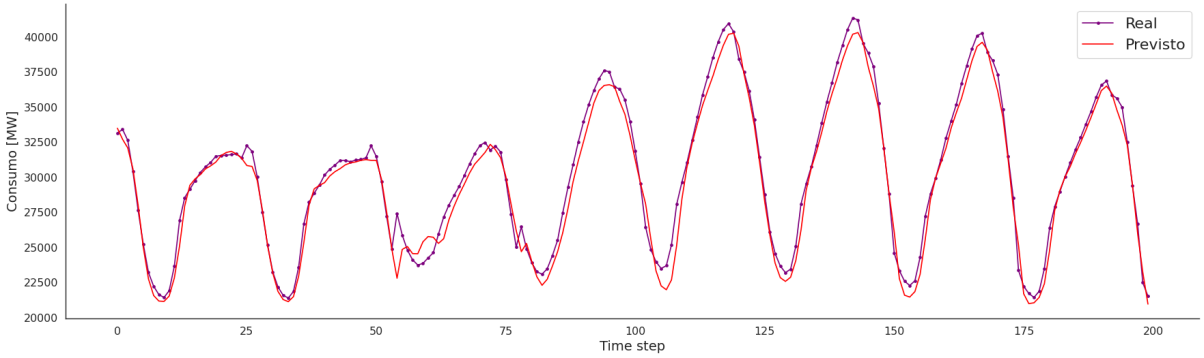
4.5 CAMADA LSTM COM FUNÇÃO DE ATIVAÇÃO RELU E 80% DO *DATASET* PARA TREINO E 20% PARA TESTE

Figura 8 - Perda do modelo com uma camada LSTM, função de ativação ReLU e 80% do *dataset* para treino e 20% para teste usado.

a) Perda do modelo.



b) Comparação dos resultados obtidos com os reais



Fonte: Autoria própria

O modelo faz uso de uma camada LSTM e função de ativação ReLU, tendo como objetivo introduzir não linearidades na camada LSTM, permitindo assim que o modelo aprendesse relações complexas entre as variáveis de entrada e saída.

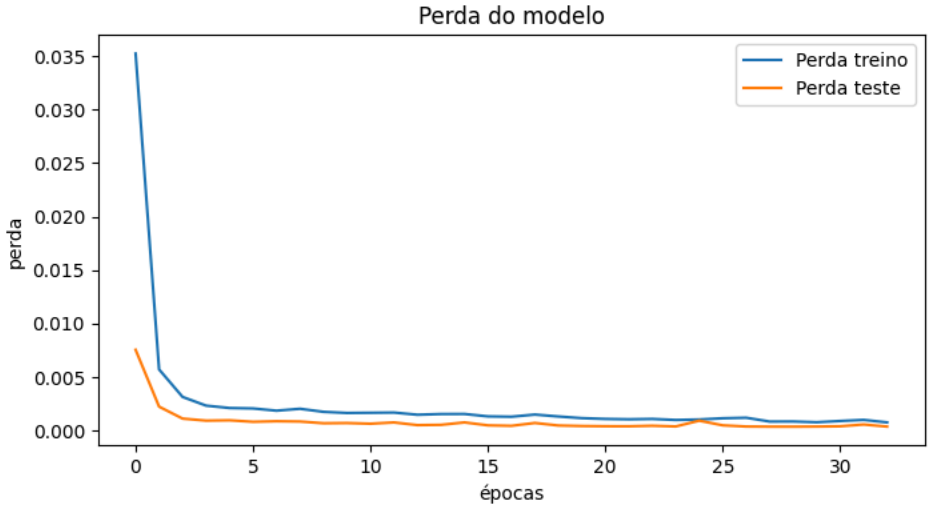
Um modelo LSTM com uma camada de ativação ReLU tem o potencial de capturar padrões complexos nos dados de consumo de energia e fazer previsões precisas no conjunto

de dados. O que o modelo apresenta, seguindo de perto com precisão os dados reais, contudo, é uma dificuldade em responder em alterações únicas de um determinado espaço.

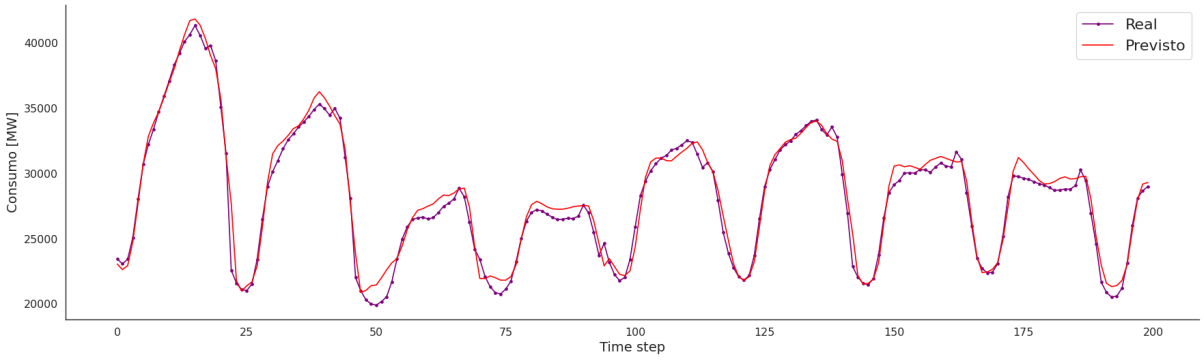
4.6 CAMADA LSTM COM FUNÇÃO DE ATIVAÇÃO RELU E 70% DO *DATASET* PARA TREINO E 30% PARA TESTE

Figura 9 - Perda do modelo com uma camada LSTM, função de ativação ReLU e 70% do *dataset* para treino e 30% para teste usado.

a) Perda do modelo.



b) Comparação dos resultados obtidos com os reais.



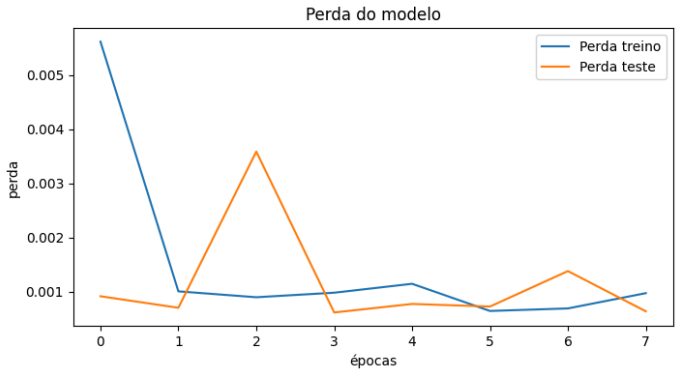
Fonte: Autoria própria

Novamente, pode se apontar que o modelo apresenta maior generalidade quando os conjuntos de dados são reduzidos, a custo de precisão. Além disso, é possível notar também que a perda em conjunto de teste e treino tenha tendência a se manter afastada por mais algumas épocas, diferentemente das versões anteriores desse mesmo modelo, que estavam mais próximas de convergir.

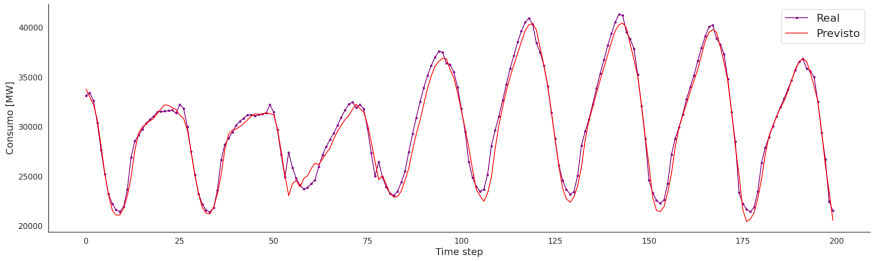
4.7 GRU BIDIRECIONAL E 80% DO *DATASET* PARA TREINO E 20% PARA TESTE

Figura 10 - Perda do modelo com uma camada GRU Bidirecional e 80% do *dataset* para treino e 20% para teste

a) Perda do modelo.



b) Comparação dos resultados obtidos com os reais



Fonte: Autoria própria

Os modelos LSTM e os modelos GRU bidirecionais representam duas abordagens distintas na modelagem de sequências. Enquanto os LSTMs são mais complexos e capazes de capturar dependências de longo prazo em sequências de dados, os GRUs são mais simples e eficientes, sendo adequados para tarefas que exijam aprendizado de dependências de curto e médio prazo. A bidirecionalidade dos modelos GRU permite a captura de informações de contexto tanto do passado quanto do futuro, proporcionando uma compreensão mais abrangente das sequências. A escolha entre os dois modelos depende das necessidades específicas do problema, do tamanho do conjunto de dados e dos recursos computacionais disponíveis. Em resumo, os LSTMs oferecem poder e flexibilidade, enquanto os modelos GRU bidirecionais são mais simples e eficientes, adaptando-se a uma variedade de aplicações de modelagem de sequências.

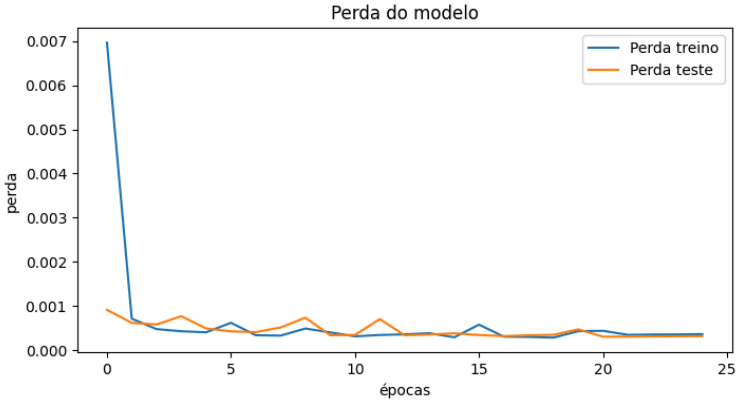
O modelo GRU conseguiu responder com certa precisão para rápidas variações. Contudo quando o modelo precisava entregar menores valores, não apresentou bons

resultados, mesmo com os conjuntos de perda de teste e treino convergindo para baixos valores, inclusive apresentando neste picos inesperados.

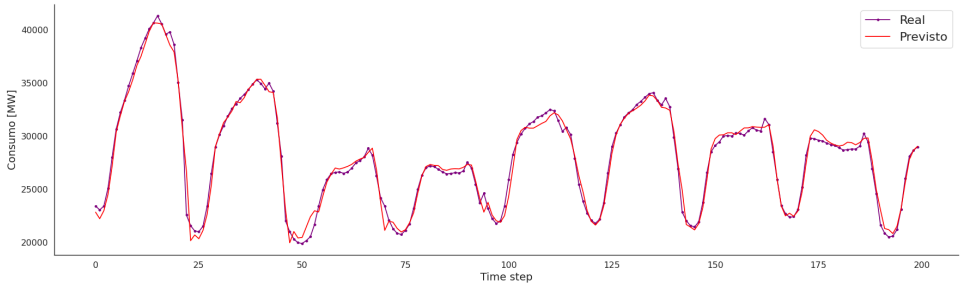
4.8 GRU BIDIRECIONAL E 70% DO *DATASET* PARA TREINO E 30% PARA TESTE

Figura 11 - Perda do modelo com uma camada GRU Bidirecional e 70% do *dataset* para treino e 30% para teste usado.

a) Perda do modelo.



b) Comparação dos resultados obtidos com os reais.



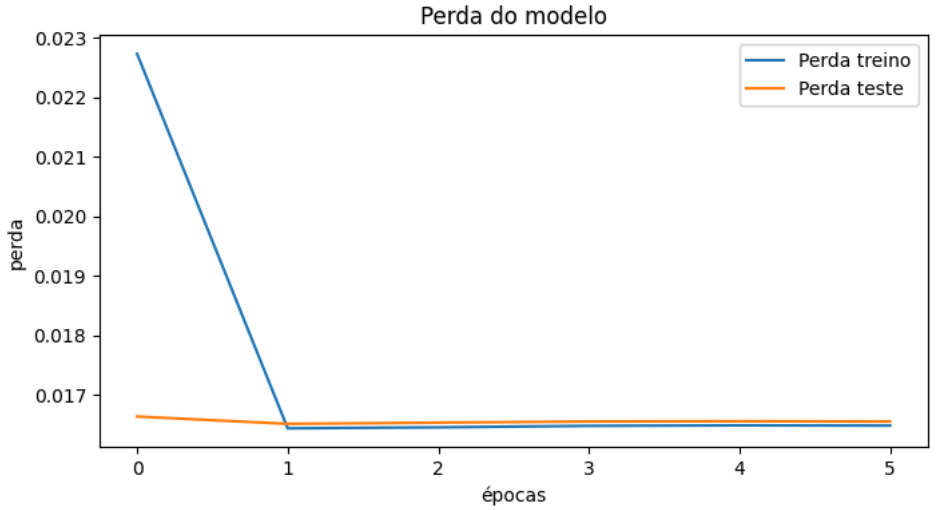
Fonte: Autoria própria

Ao se substituir o conjunto de dados para 70% do *dataset* original, as previsões feitas pelo modelo diminuíram muito em precisão, além de a perda do modelo ser conflituosa, se mostrando menos eficaz que a versão com 80% do *dataset*.

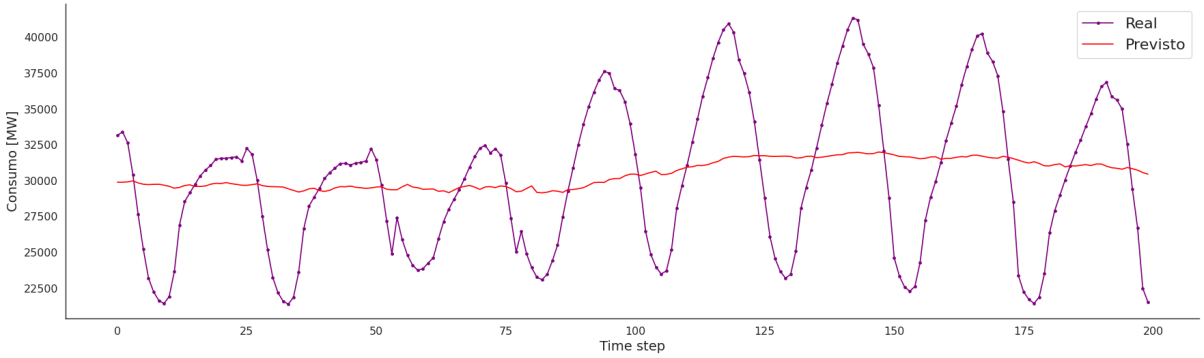
4.9 GRU COM CAMADA DE ATENÇÃO E 80% DO *DATASET* PARA TREINO E 20% PARA TESTE

Figura 12 - Perda do modelo com uma camada GRU com camada de atenção e 80% do *dataset* para treino e 20% para teste usado.

a) Perda do modelo.



b) Comparação dos resultados obtidos com os reais



Fonte: Autoria própria

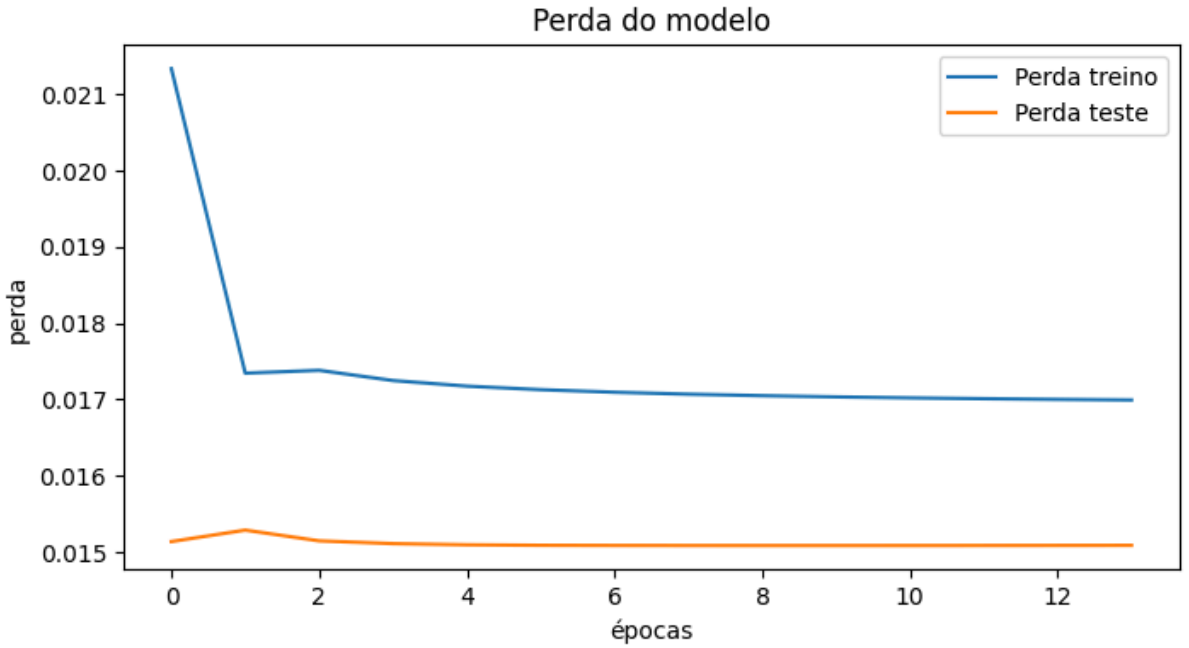
A camada de atenção em um modelo serve para permitir que o modelo pondere dinamicamente diferentes partes dos dados de entrada, destacando as partes mais relevantes para a tarefa em questão e melhorando a capacidade do modelo de lidar com sequências e contextos complexos.

Porém, devido ao formato dos dados, o modelo não foi capaz de sensibilizar suficientemente os dados, prejudicando de forma extrema o resultado final. É possível notar que quando a amplitude muda, as predições até acompanham minimamente a região de média dos dados reais, porém a enorme discrepância entre os resultados obtidos e esperados o torna completamente inadequado.

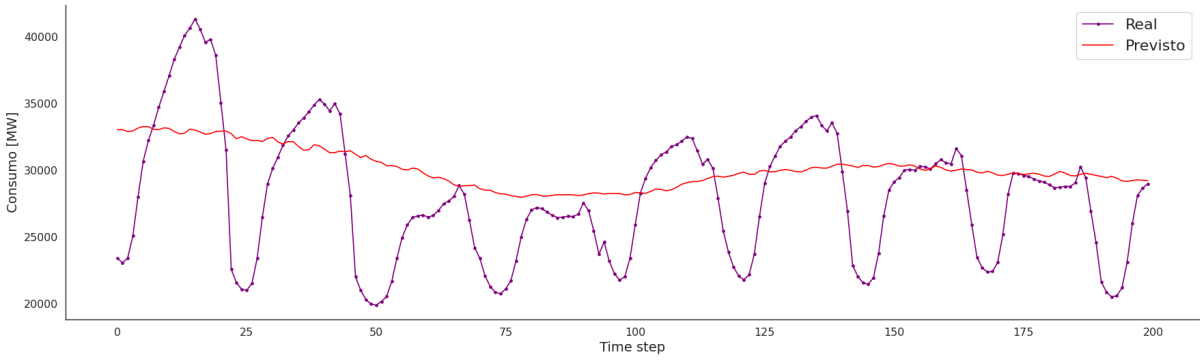
4.10 GRU COM CAMADA DE ATENÇÃO E 70% DO *DATASET* PARA TREINO E 30% PARA TESTE

Figura 13 - Perda do modelo com uma camada GRU com camada de atenção e 70% do *dataset* para treino e 30% para teste usado.

a) Perda do modelo.



b) Comparação dos resultados obtidos com os reais



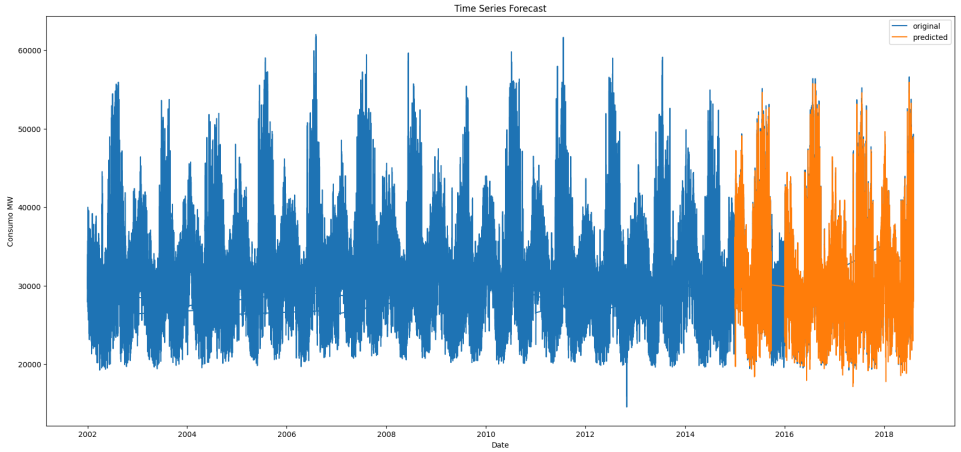
Fonte: Autoria própria

Novamente, o modelo GRU com camada de atenção se mostra ineficiente para a situação problema, sendo incapaz de realizar previsões adequadas. Além disso, ao se usar 70% dos dados, nem mesmo a perda do modelo se mostrou adequada, não convergindo em nenhum momento.

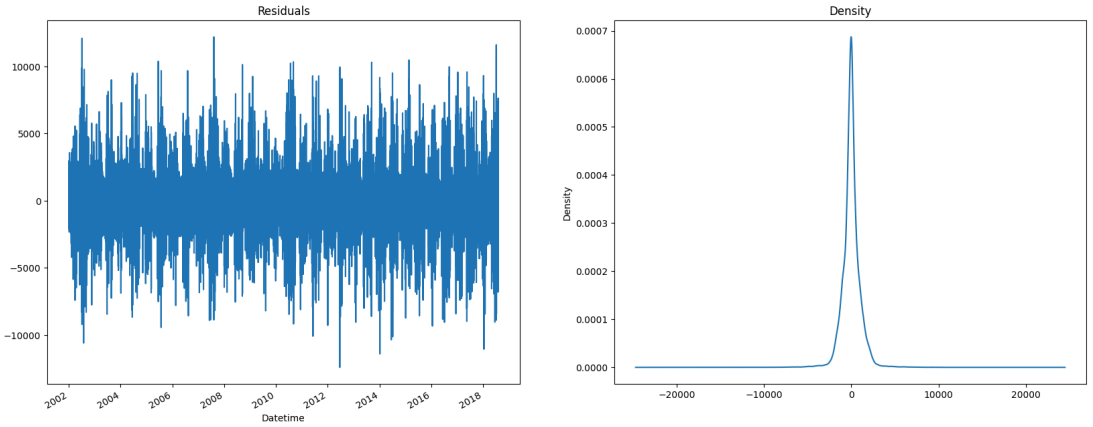
4.11 ARIMA (2,0,4) E 80% DO *DATASET* PARA TREINO E 20% PARA TESTE

Figura 14 - Representação gráfica das predições do modelo ARIMA (2,0,4) e 80% do *dataset* para treino e 20% para teste em comparação com o *dataset* real.

a) Perda do modelo.



b) Representação gráfica dos efeitos residuais e densidade



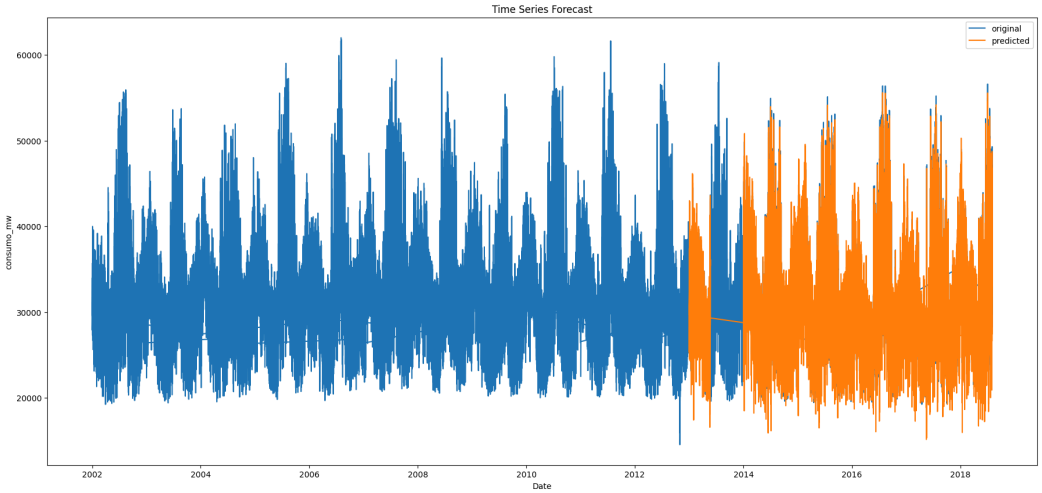
Fonte: Autoria própria

A representação gráfica dos resíduos de um modelo ARIMA é uma ferramenta valiosa para avaliar a adequação do modelo aos dados e identificar áreas onde o modelo pode precisar de ajustes ou melhorias. Ele ajuda a garantir que as suposições do modelo ARIMA sejam atendidas e que os resíduos se comportam como esperado. Além disso, um pico de densidade em torno de zero nos resíduos de um modelo ARIMA é um sinal positivo e sugere que o modelo está bem ajustado aos dados, produzindo previsões estáveis e confiáveis. Foram usados esses métodos gráficos apesar da diferença com os dos modelos LSTM e GRU apenas por uma questão de estrutura de modelo. Ainda serão aplicados aos modelos as mesmas métricas que permitirão determinar seu desempenho.

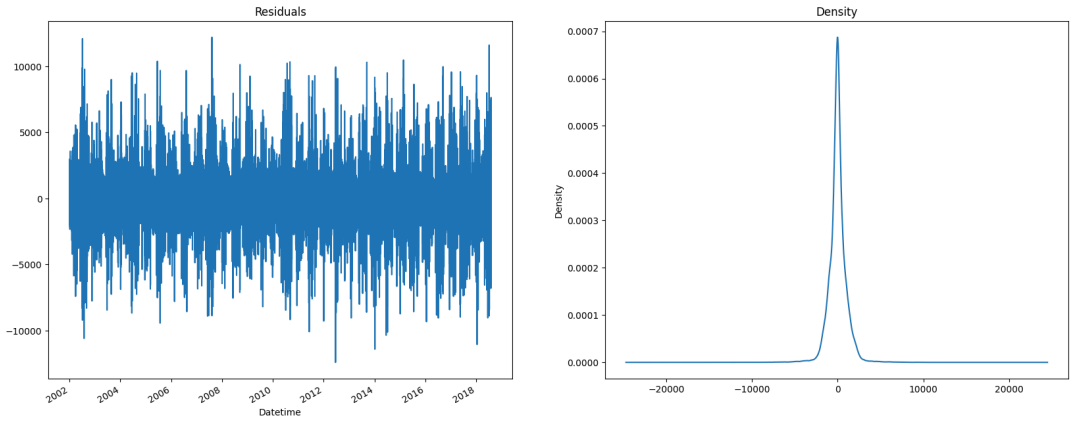
4.12 ARIMA (2,0,4) E 70% DO *DATASET* PARA TREINO E 30% PARA TESTE

Figura 15 - Representação gráfica das predições do modelo ARIMA (2,0,4) e 70% do *dataset* para treino e 30% para teste em comparação com o *dataset* real.

a) Perda do modelo.



b) Representação gráfica dos efeitos residuais e densidade



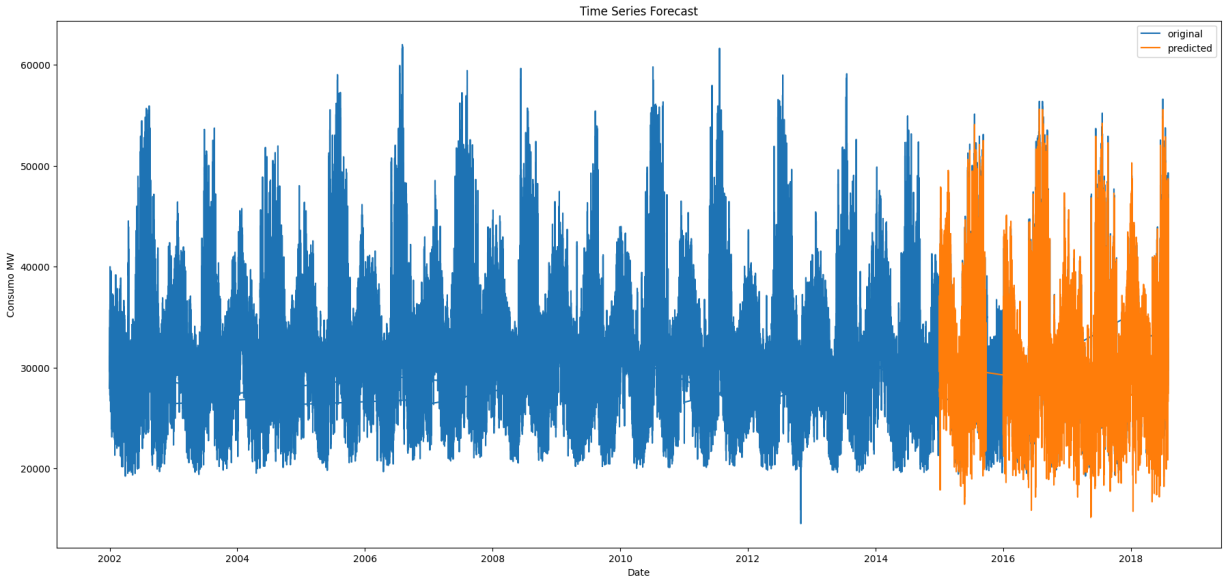
Fonte: Autoria própria

Similar ao modelo anterior, não é possível determinar grandes diferenças entre os modelos, mas aparentemente apresenta bons resultados e predições.

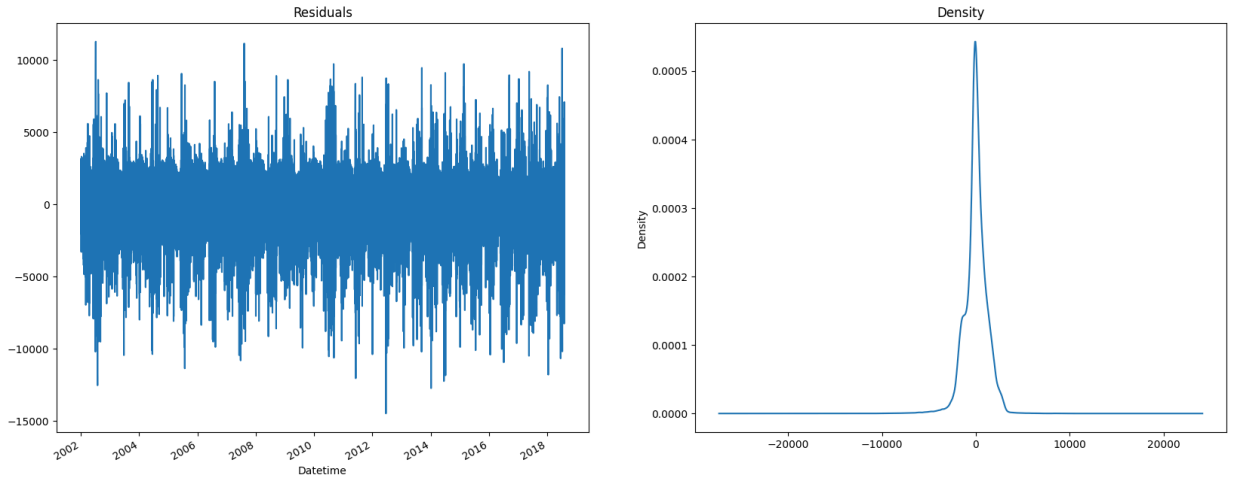
4.13 ARIMA (1,0,1) E 80% DO *DATASET* PARA TREINO E 20% PARA TESTE

Figura 16 - Representação gráfica das predições do modelo ARIMA (1,0,1) e 80% do *dataset* para treino e 20% para teste em comparação com o *dataset* real.

a) Perda do modelo.



b) Representação gráfica dos efeitos residuais e densidade



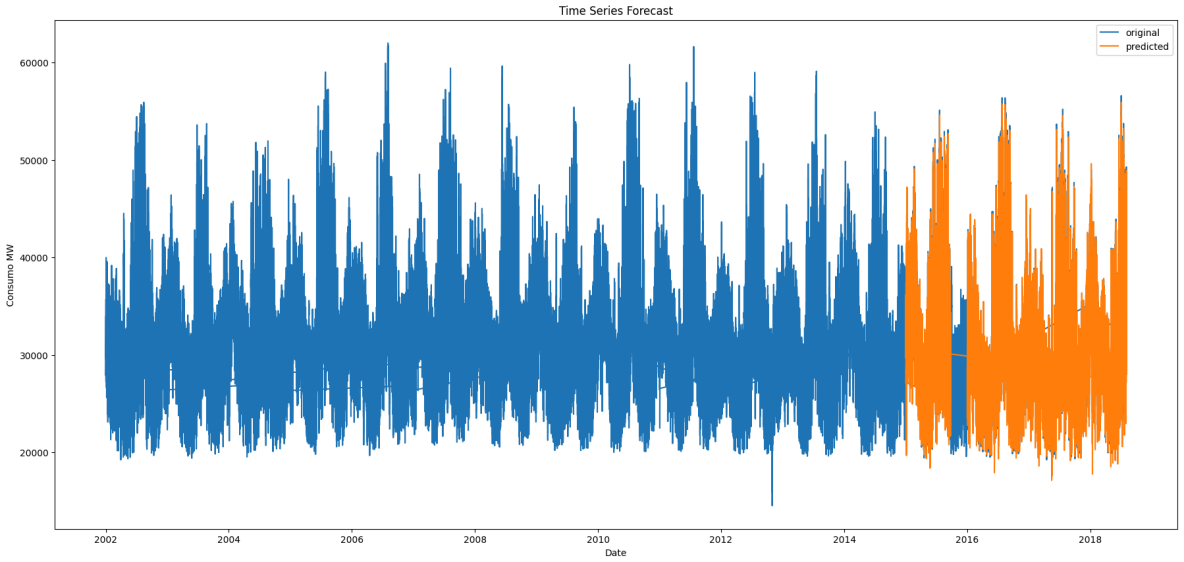
Fonte: Autoria própria

Diferentemente dos valores obtidos para os modelos das duas seções imediatamente anteriores, há distorção tanto na densidade e picos em seus residuais, o que evidencia que os parâmetros para a ordem do modelo, o grau de diferenciação e ordem do modelo de média móvel não estão adequados.

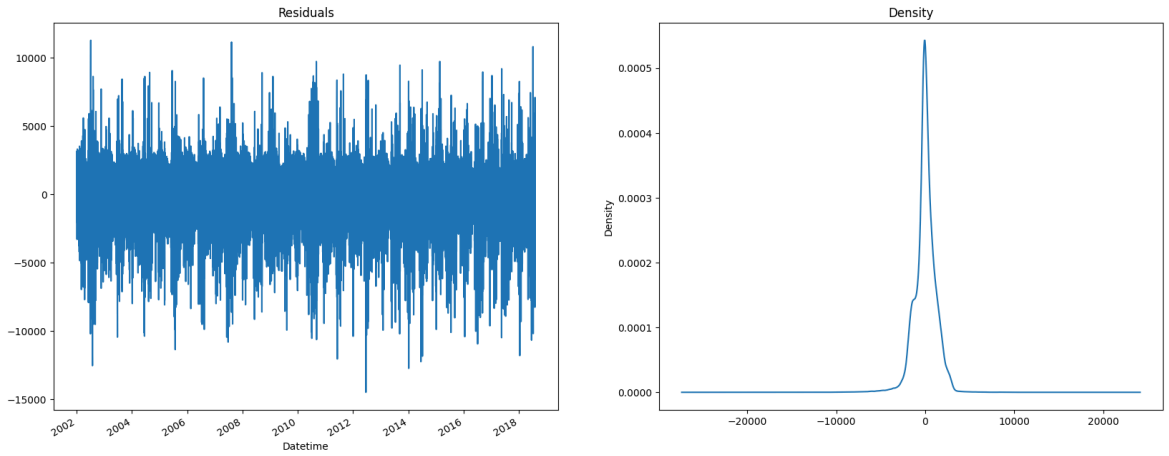
4.14 ARIMA (1,0,1) E 70% DO *DATASET* PARA TREINO E 30% PARA TESTE

Figura 17 - Representação gráfica das predições do modelo ARIMA (1,0,1) e 70% do *dataset* para treino e 30% para teste em comparação com o *dataset* real.

a) Perda do modelo.



b) Representação gráfica dos efeitos residuais e densidade



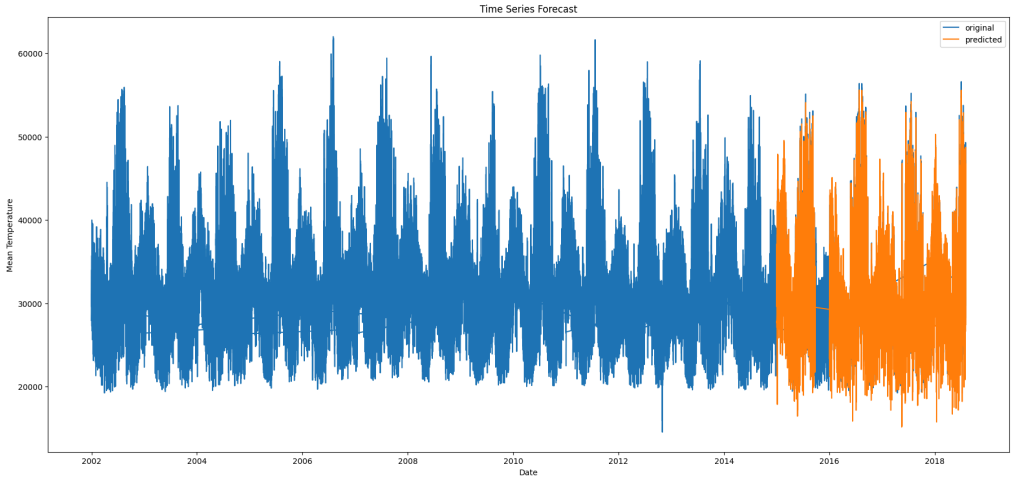
Fonte: Autoria própria

Novamente, fica claro que alterar o dimensionamento do conjunto de teste e treino apresenta poucas discrepâncias no modelo ARIMA, de forma que é de fato necessário ajustar a ordem do modelo, o grau de diferenciação e ordem do modelo de média móvel para que o modelo projetado se torne de fato adequado à situação de uso.

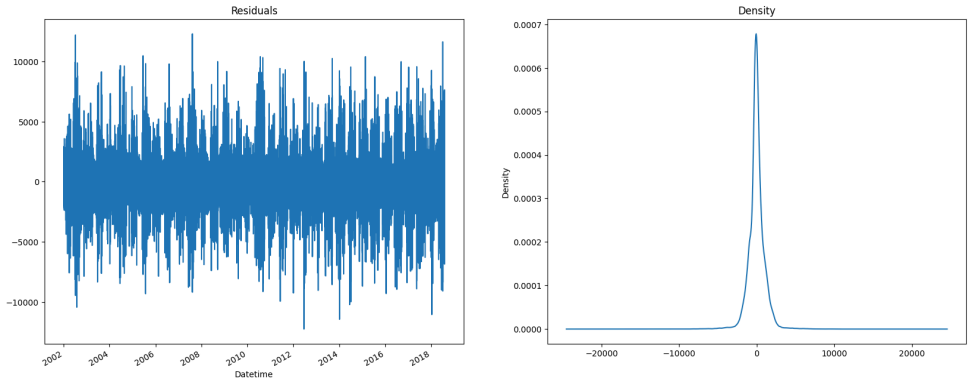
4.15 ARIMA (3,0,3) E 80% DO *DATASET* PARA TREINO E 20% PARA TESTE

Figura 18 - Representação gráfica das predições do modelo ARIMA (3,0,3) e 80% do *dataset* para treino e 20% para teste em comparação com o *dataset* real.

a) Perda do modelo.



b) Representação gráfica dos efeitos residuais e densidade



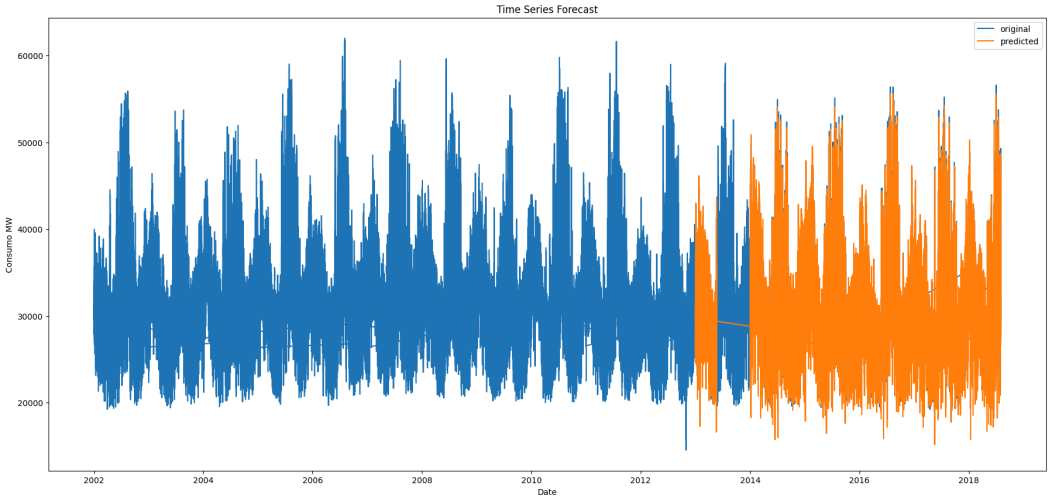
Fonte: Autoria própria

Apresenta resultados extremamente similares aqueles obtidos pelo modelo ARIMA (2,0,4), devido a proximidade dos valores da ordem do modelo, o grau de diferenciação e ordem do modelo de média móvel, porém aparenta apresentar menores valores de pico para os resíduos.

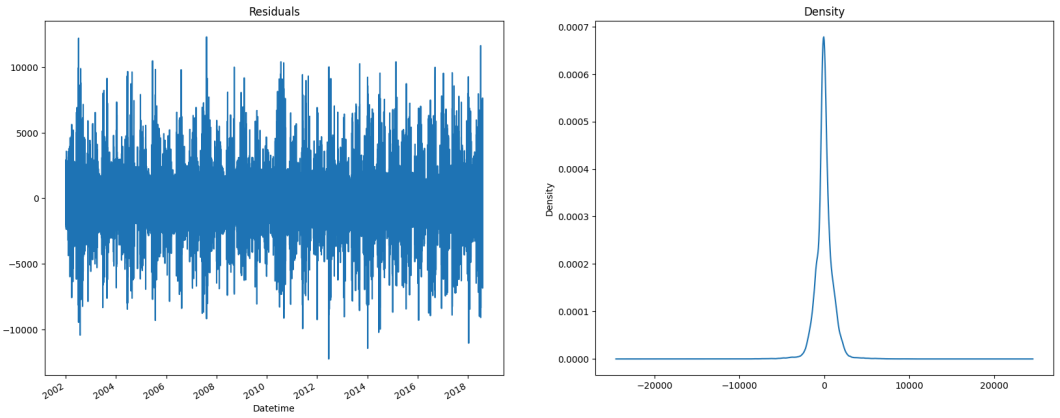
4.16 ARIMA (3,0,3) E 70% DO *DATASET* PARA TREINO E 30% PARA TESTE

Figura 19 - Representação gráfica das predições do modelo ARIMA (3,0,3) e 70% do *dataset* para treino e 30% para teste em comparação com o *dataset* real.

a) Perda do modelo.



b) Representação gráfica dos efeitos residuais e densidade



Fonte: Autoria própria

Por fim, o último modelo executado reforça o que foi comprovado anteriormente, que de fato as alterações em um modelo ARIMA são somente visíveis quando se altera a ordem do modelo, o grau de diferenciação e ordem do modelo de média móvel.

4.17 RESULTADOS FINAIS CONSOLIDADOS

Obtidos os resultados para cada métrica previamente estabelecida, foi possível então os consolidar e comparar, de forma a tornar possível determinar qual deles foi o mais performático do conjunto. Além disso, para que fosse possível estabelecer comparações coerentes, as métricas obtidas foram separadas em dois conjuntos de dados, o primeiro sendo os resultados provenientes da divisão do *dataset* original, conforme mencionado na seção 3.3.

4.17.1 DATASET *SEPARADO EM 80% DE TREINAMENTO E 20% DE TESTE*

TABELA 3 - Métricas de avaliação para os modelos com *dataset* dividido em 80% de treinamento e 20% de teste.

Modelo	Camada LSTM	Três camadas LSTM	Função de ativação reLu	GRU Bidirecional	GRU com camada de atenção	ARIMA 2,0,4	ARIMA 1,0,1	ARIMA 3,0,3
MAE treino [MW]	670,27	590,87	713,81	607,46	4.197,21	883,04	519,35	868,55
RMSE treino [MW]2	951,88	868,65	987,71	850,87	5.307,63	1.184,74	717,90	1.175,53
DTW treino	324.563,77	296.184,45	336.778,91	290.122,85	1.809.746,54	404.016,77	244.815,05	400.874,39
SMAPE treino	1,80E-05	1,83E-05	1,92E-05	1,61E-05	1,14E-04	7,39E-05	9,30E-05	7,43E-05
MAE teste [MW]	680,16	616,58	705,59	611,73	4.017,08	679,90	854,84	683,13
RMSE teste [MW]2	987,15	920,78	1.011,94	896,07	5.212,25	1.061,88	1.232,68	1.066,28
DTW teste	168.230,77	156.919,57	172.454,56	152.707,88	888.271,36	181.062,51	210.185,01	181.813,18
SMAPE teste	7,45E-05	7,76E-05	7,73E-05	6,63E-05	4,43E-04	2,36E-05	1,39E-05	2,32E-05

Fonte: Autoria própria

Algumas hipóteses que foram levantadas ao se analisar apenas os resultados expressos graficamente se tornam evidentes ao se colocar os resultados das métricas lado a lado. O primeiro ponto de grande evidência é que, de fato, o modelo com piores resultados e maiores

erros é o modelo GRU com camada de atenção. Além disso, para determinar o modelo que melhor performou, é necessário olhar não só aquele que possui menores valores para os erros, mas sim o que apresenta menores valores de métricas mais vezes, possuindo portanto uma menor variação. Dito isso, o modelo que possui melhor performance é o Três camadas LSTM.

O fato de que o melhor resultado obtido foi através de um modelo com três camadas LSTM aponta para o alto grau de complexidade da situação que este estudo tenta resolver, de forma que múltiplas camadas são necessárias para capturar adequadamente os padrões nos dados. Problemas com características temporais complexas, dependências de longo prazo ou padrões sutis podem se beneficiar de modelos mais profundos. Os modelos com múltiplas camadas LSTM têm uma capacidade de aprendizado mais expressiva em comparação com modelos mais simples. Isso significa que eles são capazes de aprender representações mais complexas e capturar relações mais profundas nos dados, o que pode resultar em previsões mais precisas e generalizáveis. No entanto, modelos mais complexos, como aqueles com múltiplas camadas LSTM, têm maior probabilidade de sofrer de overfitting, especialmente se o conjunto de dados não for grande o suficiente ou se não houver técnicas de regularização adequadas implementadas, como foi levantado na seção anterior.

4.17.2 DATASET *SEPARADO EM 70% DE TREINAMENTO E 30% DE TESTE*

A partir dos dados obtidos para o conjunto de treino e teste com a divisão de 70% e 30%, respectivamente, foi possível criar a tabela abaixo, contendo os valores para as métricas mencionadas previamente.

Novamente, o pior resultado obtido foi para o modelo GRU com camada de atenção, o que era esperado, como visto e analisado anteriormente no estudo da representação gráfica dos resultados obtidos.

No entanto, o modelo com uma única camada LSTM obteve o melhor desempenho. Uma divisão diferente dos dados em relação ao *dataset* anterior sugere que a escolha da arquitetura do modelo e a distribuição dos dados entre conjuntos de treinamento e teste são fatores críticos a serem considerados na construção de modelos de previsão. Experimentar com diferentes arquiteturas e divisões de dados pode ajudar a entender melhor como esses fatores impactam o desempenho do modelo e sua capacidade de generalização para novos dados. O fato do modelo com apenas uma camada ter sido o que melhor performou não é surge como uma grande surpresa, uma vez que durante a análise prévia este foi o que menos

generalizou ao se mudar o *dataset* utilizado. Apesar de modelos como ARIMA mostrarem bons resultados, para erro quadrado absoluto, DTW e erro médio quadrado, deve-se levar em consideração os resíduos e densidade, o que faz com que eles percam em critérios de avaliação quando comparados aos demais modelos.

TABELA 4 - Métricas de avaliação para os modelos com *dataset* dividido em 70% de treinamento e 30% de teste.

Modelo	Camada LSTM	Três camadas LSTM	Função de ativação reLu	GRU Bidirecional	GRU com camada de atenção	ARIMA 2/0/4	ARIMA 3/0/3	ARIMA 1/0/1
MAE treino [MW]	647,29	737,85	721,39	799,85	4.178,90	894,24	879,42	519,35
RMSE treino [MW]²	938,45	986,76	985,57	1.002,88	5.319,58	1.198,36	1.189,27	717,90
DTW treino	299.314,13	314.720,88	314.342,19	319.861,77	1.696.646,53	382.268,36	379.367,35	244.815,05
SMAPE treino	1,97 E-05	1,74E-05	2,18E-05	2,40E-05	1,28E-04	4,95E-05	4,97 E-05	9,30E-05
MAE teste [MW]	652,01	764,94	767,83	824,80	3.846,26	675,931	678,86	854,83
RMSE teste [MW]²	961,99	1.027,22	1.039,32	1.050,28	4.974,47	1046,88	1050,78	1232,67
DTW teste	200.821,46	214.437,72	216.965,06	219.251,14	1.038.450,04	218620,92	219434,72	210185,00
SMAPE teste	4,78E-05	4,32E-05	5,59E-05	5,98E-05	2,82E-04	2,71E-05	2,67E-05	1,39E-05

Fonte: Autoria própria

5 CONCLUSÃO

O uso de modelos preditivos baseados em séries temporais vem sendo difundido e amplamente utilizado em uma variedade de campos e setores em todo o mundo. Apesar do avanço de outras técnicas de aprendizado de máquina, como redes neurais profundas e algoritmos de árvores de decisão, os métodos de previsão de séries temporais ainda são altamente relevantes e eficazes em muitas aplicações. Tomando por exemplo a média de temperatura global, o uso de uma rede neural baseada em séries temporais pode ser obter o quanto a temperatura irá subir ou cair em um determinado período de tempo no futuro, e, com os dados de temperatura desde a primeira revolução industrial, por exemplo, determinar a influência humana nesse período, ou até mesmo criar um modelo com dados prévios a primeira revolução industrial e assim determinar o quanto a temperatura deveria estar com menor interferência humana.

O estudo aqui realizado teve como objetivo desenvolver um modelo para determinação de consumo energético com foco na precisão dos resultados, visto que atualmente a demanda mundial tem aumentado e este se trata de um fator que apresenta um risco imenso para a população mundial e para o planeta. A análise dos resultados obtidos revelou que, para o modelo que faz uso de três camadas LSTM, o erro médio absoluto foi de 590,87 MW, o erro rms quadrado 868,6 MW e um DTW de 296.184,45. Todos esses valores se mostraram sólidos demonstrando a precisão do modelo aqui desenvolvido. Além disso, os Apêndices A e B mostram graficamente a diferença entre os resultados obtidos através das previsões e valores reais.

Ao se estressar essa topologia e a comparar com outras em diversos cenários, os modelos LSTM ainda se mostraram mais efetivos, o que ilustra o caminho para pesquisas mais aprofundadas sobre o tópico, permitindo assim que próximos modelos desenvolvidos sejam cada vez mais precisos, elemento este que se mostra fundamental dada a complexidade do problema apresentado.

Podem ser mencionados também os resultados presentes nas comunidades online, já citadas anteriormente, onde também foram desenvolvidas estruturas LSTM e GRU. Devido a diferença de ajustes nos parâmetros, houve variação nos resultados obtidos, como é possível estabelecer comparando o valor de SMAPE obtido no melhor modelo aqui essa comparação

entre GRU e LSTM[10], de 0.261%. Grandes diferenças de parâmetros e diferença nas métricas utilizadas para comparação tornam uma comparação mais profunda levemente inviável, porém abre espaço para demais estudos que podem ser levantados a partir dos modelos aqui desenvolvidos, observando outros *datasets*, em especial os brasileiros, trazendo essa ótica para dentro do país, avançando esse tópico cada vez mais, trazendo ele para o dia a dia e difundindo conhecimento. Uma sugestão é o *dataset* Consumo Mensal de Energia Elétrica por Classe (regiões e subsistemas)[11], disponibilizado pela Empresa de Pesquisa Energética (EPE). Apesar da frequência de coleta de dados não ser tão grande quanto a aqui utilizada, pode ser um ótimo início para pesquisas brasileiras. Outra metodologia que pode ser aplicada em busca de melhorias, seria a Box-Jenkins nos modelos ARIMA, em busca de fornecer uma estrutura sistemática e rigorosa para a identificação, estimação e diagnóstico do modelo, obtendo assim melhores resultados.

Dessa forma, é crucial manter um horizonte de continuidade, visando o desenvolvimento de projetos cada vez mais importantes para nossa sociedade e planeta. Para avançar, é necessário explorar novas metodologias e topologias, como as *liquid nets* e *transformers*, que representam abordagens inovadoras no campo da modelagem preditiva. Além disso, a utilização de dados de alta-frequência oferece uma oportunidade significativa para aprimorar a precisão e a abrangência das análises e previsões realizadas, permitindo assim uma compreensão mais detalhada e dinâmica dos padrões de consumo energético.

[10] Disponível em: https://www.kaggle.com/code/gabrielloye/gru-vs-lstm-prediction. Acesso em: 30 fev. 2024

[11] Disponível em: https://www.epe.gov.br/sites-pt/publicacoes-dados-abertos/publicacoes/Documents/CONSUMO%20MENSAL%20DE%20ENERGIA%20EL%C3%89TRICA%20POR%20CLASSE.xls
Acesso em: 30 fev. 2024

REFERÊNCIAS

ARAÚJO, Eduardo Gomes de. **Modelagem e previsão de modelos de séries temporais do consumo de energia elétrica na região Nordeste do Brasil**. Universidade Estadual da Paraíba, Centro de Ciências e Tecnologia , 2021. Acesso em: 24 fev. 2024. Disponível em: https://dspace.bc.uepb.edu.br/jspui/bitstream/123456789/25769/1/PDF%20-%20Eduardo%20Gomes%20de%20Araújo

BALOGLU, Orkun *et al*. **What is machine learning?** Arch Dis Child Educ Pract, vol. 107, ed. 5, pp.: 386-388, 2022. Disponível em: https://pubmed.ncbi.nlm.nih.gov/33558304/ . Acesso em: 24 fev. 2024.

BARROS, Lutero Paes de. **Modelo preditivo do consumo de energia elétrica utilizando séries temporais, para avaliar a influência do sistema de geração distribuída em Mato Grosso, iniciado em 2016**. 2022. Dissertação (Mestrado em Ciências Ambientais) – Universidade de Cuiabá, Cuiabá, 2022. Disponível em: https://repositorio.pgsscogna.com.br/bitstream/123456789/53108/1/Dissertacao_Mestrado_Luter o_Final.pdf.. Acesso em: 24 fev. 2024.

BENGIO, Yoshua. **Learning Deep Architectures for AI.** Foundations and Trends in Machine Learning, vol. 2, ed. 1, pp.: 1-127, 2009. Disponível em: https://www.nowpublishers.com/article/Details/MAL-006 . Acesso em: 24 fev. 2024.

BERRIEL, Rodrigo *et al*. Monthly energy consumption forecast: A deep learning approach. pp 4283-4290, 2017. 10.1109/IJCNN.2017.7966398. Disponível em:https://www.researchgate.net/publication/318332587_Monthly_energy_consumption_forecast_A_deep_learning_approach. Acesso em: 30 fev. 2024.

BISHOP, C. M. **Pattern Recognition and Machine Learning**. Springer, pp, 255-284, 2006 Disponível em: https://www.microsoft.com/en-us/research/uploads/prod/2006/01/Bishop-Pattern-Recognition-and-Machine-Learning-2006.pdf Acesso em: 30 fev. 2024.

BOX, George *et al*. **Time series analysis: forecasting and control.** John Wiley & Sons. Nova Jersey, 2015.

DAVID, Rolnik *et al*. **Tackling Climate Change with Machine Learning**. ACM Comput. Surv. vol. 55, ed. 2, art. 42, 2023. Disponível em: https://dl.acm.org/doi/10.1145/3485128 . Acesso em: 24 fev. 2024.

DELIANIDI, Marina *et a*l. **KT-Bi-GRU: Student Performance Prediction with a Bi-Directional Recurrent Knowledge Tracing Neural Network.** Journal of Educational Data Mining, vol. 15, ed. 2, pp.:1–21, 2023. Disponível em: https://doi.org/10.5281/zenodo.7808087 . Acesso em: 24 fev. 2024.

HOCHREITER, Sepp *et al*. **Long Short Term Memory.** Neural Computation, vol. 9, ed. 8, pp. 1750, 1997. Disponível em https://www.researchgate.net/publication/13853244_Long_Short-term_Memory . Acesso em: 24 fev. 2024.

HOSSAIN, Eklas *et al*. **Application of Big Data and Machine Learning in Smart Grid, and Associated Security Concerns: A Review.** IEEE Access, vol. 7, pp.: 13960-13988, 2019. Disponível em: https://ieeexplore.ieee.org/document/8625421 . Acesso em: 24 fev. 2024.

HYNDMAN, Rob, ATHANASOPOULOS, George. **Forecasting: Principles and practice. OTexts**. Disponível em: https://otexts.com/fpp3/. Acesso em: 24 fev. 2024.

IFTIKHAR, Hasnain, *et al*. 2023. **"Multiple Novel Decomposition Techniques for Time Series Forecasting: Application to Monthly Forecasting of Electricity Consumption in Pakistan"** Energies 16, no. 6: 2579. Acesso em: 24 fev. 2024. Disponível em: https://doi.org/10.3390/en16062579

KHAN, Prince Waqas *et al*. **Machine Learning-Based Approach to Predict Energy Consumption of Renewable and Non Renewable Power Sources**. Energies, vol. 13, pp.: 4870, 2020. Disponível em: https://www.mdpi.com/1996-1073/13/18/4870. Acesso em: 24 fev. 2024.

KYUNGHUN, Cho *et al*. **Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation.** 2014. Disponível em: https://arxiv.org/abs/1406.1078 . Acesso em: 24 fev. 2024.

LI, Bei *et al*. **Predicting user comfort level using machine learning for smart grid environments.** IEEE PES Innov. Smart Grid Technol. (ISGT), 2011. Disponível em: https://www.researchgate.net/publication/224233986_Predicting_user_comfort_level_using_machine_learning_for_Smart_Grid_environments . Acesso em: 24 fev. 2024.

LIU, Yun *et al*. **Machine learning for advanced energy materials**, Energy and AI, vol. 3, 2021. Disponível em: https://www.sciencedirect.com/science/article/pii/S2666546821000033. Acesso em: 24 fev. 2024.

MATHEW, MD. **Nuclear energy: A pathway towards mitigation of global warming.** Progress in Nuclear Energy, vol. 143, 2022. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S0149197021004340. Acesso em: 24 fev. 2024.

MOSAVI, Amir *et al*. **State of the Art of Machine Learning Models in Energy Systems, a Systematic Review.** Energies, vol. 12, pp.: 1301, 2019. Disponível em: https://www.mdpi.com/1996-1073/12/7/1301. Acesso em: 24 fev. 2024.

MUSBAH, Hmeda *et al*. **Energy management of hybrid energy system sources based on machine learning classification algorithms**. Electric Power Systems Research, vol. 199, 2021. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S037877962100417X. Acesso em: 24 fev. 2024.

NIU, Zhaoyang *et al*. **A review on the attention mechanism of deep learning.** Neurocomputing, vol. 452, pp.: 48-62, 2021. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S092523122100477X. Acesso em: 24 fev. 2024.

RAFTERY, Adrian, **Time Series Analysis.** European Journal of Operational Research, vol. 20, ed. 2, pp.: 127-137, 1985. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/0377221785900529 . Acesso em: 24 fev. 2024.

REMANI, T. *et al*. **Residential Load Scheduling With Renewable Generation in the Smart Grid: A Reinforcement Learning Approach,** IEEE Systems Journal, vol. 13, ed. 3, pp.: 3283-3294, 2019. Disponível em: https://www.researchgate.net/publication/326857274_Residential_Load_Scheduling_With_Renewable_Generation_in_the_Smart_Grid_A_Reinforcement_Learning_Approach. Acesso em: 24 fev. 2024.

SILVA, Cidiney. **Predição de Séries Temporais no Contexto de Smart Grids.** Universidade Federal de Minas Gerais - UFMG. Programa de Pós-Graduação em Engenharia Elétrica – PPGEE. 2016. Disponível em: https://repositorio.ufmg.br/bitstream/1843/BUOS-APFRER/1/tese_cidiney.pdf . Acesso em: 24 fev. 2024.

SON, Namrye. "**Comparison of the Deep Learning Performance for Short-Term Power Load Forecasting**" Sustainability 13, no. 22: 12493. 2021. Disponível em: https://doi.org/10.3390/su132212493. Acesso em: 24 fev. 2024.

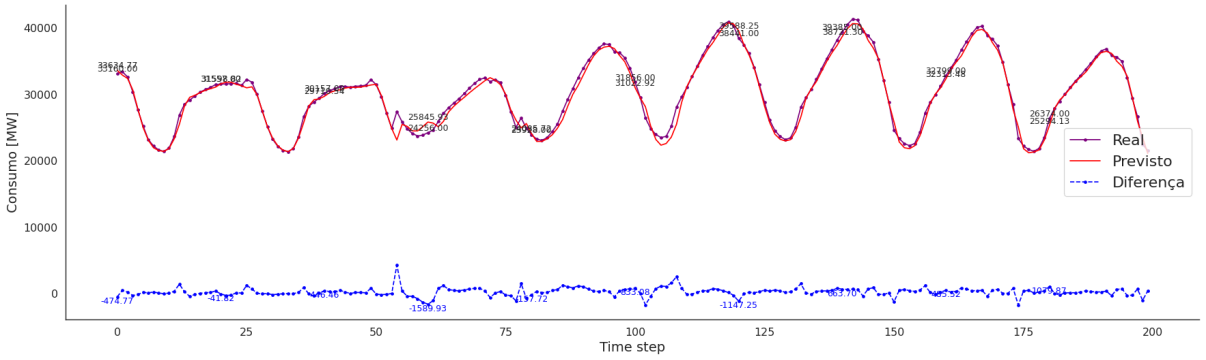
TUKEY, John. **Exploratory Data Analysis.** Pearson. Londres, 1977.

WANG, Haoxiang. **Fault Diagnosis in Hybrid Renewable Energy Sources with Machine Learning Approach.** Journal of Trends in Computer Science and Smart Technology (TCSST) vol. 3, ed. 3, pp.: 222-237, 2021. Disponível em: https://irojournals.com/tcsst/article/view/3/3/5 . Acesso em: 24 fev. 2024.

YAO, Zhenpeng *et al*. **Machine learning for a sustainable energy future**. Nat Rev Mater vol. 8, pp.: 202-215, 2022. Disponível em: https://www.nature.com/articles/s41578-022-00490-5 . Acesso em: 24 fev. 2024.

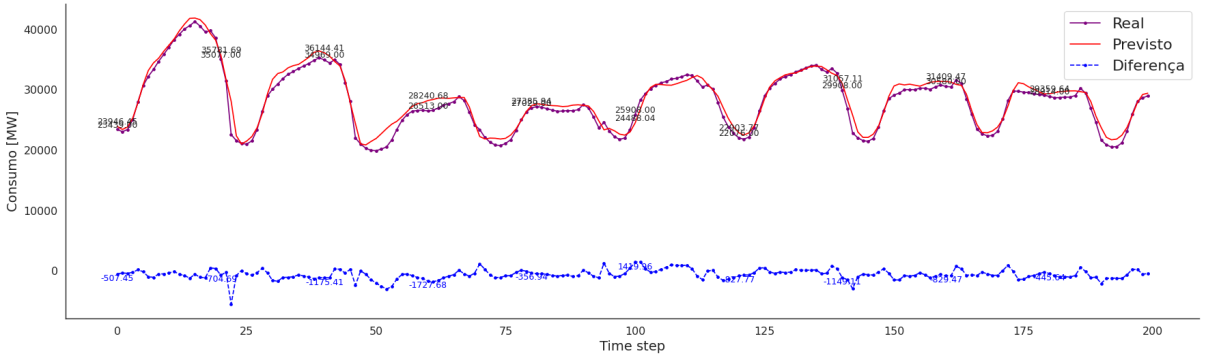
ZHANG, RUI *et al*. **Forecast of Solar Energy Production - A Deep Learning Approach,** 2018 IEEE International Conference on Big Knowledge (ICBK), Singapura, 2018, pp. 73-82, doi: 10.1109/ICBK.2018.00018. Disponível em: https://ieeexplore.ieee.org/abstract/document/8588777. Acesso em: 30 fev. 2024

APÊNDICE A - DIFERENÇA ENTRE OS RESULTADOS OBTIDOS E VALORES REAIS PARA O MODELO COM TRÊS CAMADAS LSTM E DIVISÃO DE 80% DE TREINO E 20% DE TESTE.



Fonte: Autoria própria

APÊNDICE B - DIFERENÇA ENTRE OS RESULTADOS OBTIDOS E VALORES REAIS PARA O MODELO COM UMA CAMADA LSTM E DIVISÃO DE 70% DE TREINO E 30% DE TESTE.



Fonte: Autoria própria