UNIVERSIDADE ESTADUAL PAULISTA - UNESP

Instituto de Ciência e Tecnologia- Campus de Sorocaba

FERNANDO FELIPE BAILONI ANDREONI

PREDIÇÃO DA PRODUÇÃO DE ENERGIA FOTOVOLTAICA BASEADA EM REDES NEURAIS PODADAS

Sorocaba

2026

FERNANDO FELIPE BAILONI ANDREONI

PREDIÇÃO DA PRODUÇÃO DE ENERGIA FOTOVOLTAICA BASEADA EM REDES NEURAIS PODADAS

Trabalho de Conclusão de Curso apresentado à Universidade Estadual Paulista (UNESP), Instituto de Ciência e Tecnologia, Sorocaba, para obtenção do título de Bacharel em Engenharia de Controle e Automação.

Orientador: Prof. Dr. Leopoldo André Dutra Lusquino Filho

Sorocaba
2026

A559p

Andreoni, Fernando Felipe Bailoni

Predição da produção de energia fotovoltaica baseada em redes neurais podadas / Fernando Felipe Bailoni Andreoni. -- Sorocaba, 2026

38 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: Leopoldo André Dutra Lusquino Filho

1. Energia solar. 2. Redes neurais (Computação). 3. Previsão. 4. Análise de séries temporais. 5. Compressão de dados (Computação). I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

FERNANDO FELIPE BAILONI ANDREONI

PREDIÇÃO DA PRODUÇÃO DE ENERGIA FOTOVOLTAICA BASEADA EM REDES NEURAIS PODADAS

Trabalho de Conclusão de Curso apresentado à Universidade Estadual Paulista (UNESP), Instituto de Ciência e Tecnologia, Sorocaba, para obtenção do título de Bacharel em Engenharia de Controle e Automação.

Data de defesa: 07/04/2026

BANCA EXAMINADORA

Prof. Dr. Leopoldo André Dutra Lusquino Filho
UNESP – Instituto de Ciência e Tecnologia – Campus de Sorocaba

---

Prof Dr. Eduardo Verri Liberado
UNESP – Instituto de Ciência e Tecnologia – Campus de Sorocaba

---

Prof Dr. Luis Armando De Oro Arenas
UNESP – Instituto de Ciência e Tecnologia – Campus de Sorocaba

RESUMO

A integração de fontes de energia fotovoltaica às redes elétricas convencionais exige a antecipação precisa da geração de energia para garantir o balanceamento dinâmico entre oferta e demanda. O Aprendizado de Máquina, especialmente por meio de Redes Neurais Recorrentes (RNNs) como LSTM e GRU, tem se destacado nessa tarefa de predição. Contudo, a elevada dimensionalidade e o custo paramétrico desses modelos resultam em um expressivo custo computacional. A literatura sugere que a mitigação desse custo implica indiretamente na redução do consumo energético. Este trabalho tem como objetivo desenvolver, avaliar e otimizar arquiteturas preditivas para séries temporais de geração solar, aplicando técnicas de poda (pruning) para reduzir a complexidade estrutural das redes sem comprometer sua precisão. Utilizando dados da usina fotovoltaica DKASC, modelos densos foram treinados e posteriormente submetidos a métodos de Magnitude Pruning, Clustering e Random Pruning. Os resultados indicam que a arquitetura submetida à poda por magnitude (50% de esparsidade) alcançou uma redução de aproximadamente 59% no tamanho de armazenamento (de 353 KB para 144 KB) e apresentou uma sutil melhoria na métrica sMAPE (7,46%) em comparação com o modelo não otimizado (7,58%). Adicionalmente, a técnica de clusterização demonstrou uma queda significativa no tempo de inferência (330,31 ms). Conclui-se que o excesso de parâmetros nativo às redes recorrentes pode ser suprimido, viabilizando a implementação de sistemas preditivos robustos e eficientes do ponto de vista energético.

Palavras-Chave: Energia Fotovoltaica; Redes Neurais Recorrentes; Predição de Séries Temporais; Poda de Modelos; Eficiência Computacional.

ABSTRACT

The integration of photovoltaic energy sources into conventional power grids requires accurate power generation forecasting to ensure the dynamic balance between supply and demand. Machine Learning, particularly through Recurrent Neural Networks (RNNs) such as LSTM and GRU, has emerged as a prominent tool for this prediction task. However, the high dimensionality and parametric cost of these models result in significant computational overhead. Literature suggests that mitigating this cost indirectly leads to a reduction in energy consumption. This work aims to develop, evaluate, and optimize predictive architectures for solar generation time series by applying pruning techniques to reduce the structural complexity of the networks without compromising their accuracy. Using data from the DKASC photovoltaic plant, dense models were trained and subsequently subjected to Magnitude Pruning, Clustering, and Random Pruning methods. The results indicate that the architecture subjected to magnitude pruning (50% sparsity) achieved a reduction of approximately 59% in storage size (from 353 KB to 144 KB) and showed a subtle improvement in the sMAPE metric (7.46%) compared to the non-optimized model (7.58%). Additionally, the clustering technique demonstrated a significant decrease in inference time (330.31 ms). It is concluded that the inherent parameter redundancy in recurrent networks can be suppressed, enabling the implementation of robust and energy-efficient predictive systems.

Keywords: Photovoltaic Energy; Recurrent Neural Networks; Time Series Prediction; Model Pruning; Computational Efficiency.

SUMÁRIO

1 INTRODUÇÃO

A energia solar tem surgido como uma solução limpa e sustentável, sendo uma das principais alternativas aos combustíveis fósseis. Devido a isso, ela tem atraído significativa atenção das comunidades acadêmica e industrial internacionais. O objetivo é buscar fontes de energia renovável e expandir a integração dessas fontes na rede elétrica, uma necessidade impulsionada pela urgência da crise climática atual e alinhada aos Objetivos de Desenvolvimento Sustentável (ODS) da ONU (Barhmi et al., 2024). Contudo, fontes como a solar e a eólica apresentam alta intermitência e não são previsíveis no curto prazo. Por essa razão, elas não são consideradas fontes despacháveis (ou seja, não podem ser acionadas sob demanda pelos operadores da rede). Essa limitação física torna a predição acurada da geração uma necessidade absoluta para garantir a estabilidade do sistema.

No campo da predição de energia solar, as técnicas de Aprendizado de Máquina (ML), especialmente as redes neurais recorrentes, estão substituindo rapidamente os modelos baseados em simuladores físicos. Elas oferecem soluções mais flexíveis e aplicáveis a diversos cenários das energias renováveis (Chu et al., 2024). Assim, o ML contribui positivamente para a gestão dessas fontes, auxiliando na mitigação do consumo energético global através de previsões mais acuradas. O Aprendizado de Máquina apresenta um desafio energético significativo. Redes neurais exigem um grande poder computacional para treinamento e inferência. Esse alto consumo de energia contribui para a pegada de carbono da tecnologia, contrapondo-se aos benefícios ambientais buscados pelas energias renováveis (Luers et al., 2024; Shehabi et al., 2024). Portanto, há uma dualidade clara: o ML ajuda a otimizar sistemas sustentáveis, mas, ao mesmo tempo, tem um consumo energético elevado.

Apesar disso, muitos trabalhos ainda não exploram técnicas de pruning (poda) que podem criar modelos menores e mais eficientes sem comprometer a precisão. O trabalho fundamentado por Vadera & Ameen (2022) destaca as vantagens dos métodos de poda, como magnitude-based pruning e sensitivity analysis, na otimização e simplificação das redes neurais. O desafio de lidar com a alta dimensionalidade dos dados em séries temporais é significativo e desafia arquiteturas tradicionais. Neste sentido, a aplicação de pruning promete não apenas aumentar a simplicidade e a velocidade dos modelos, mas também reduzir drasticamente seu consumo energético e o impacto ambiental associado.

Assim, a proposta deste trabalho de fim de curso é realizar predições de médio prazo da produção de energia solar (dataset DKASC) utilizando modelos de redes neurais (LSTM e GRU). O objetivo geral é aplicar técnicas de pruning (poda) nestes modelos, focando na otimização de sua eficiência energética e computacional.

Para alcançar esta meta, foram traçados os seguintes objetivos específicos:

- Avaliar e comparar o desempenho preditivo preliminar de diferentes topologias de Redes

Neurais Recorrentes (LSTM e GRU) na modelagem de séries temporais fotovoltaicas;

- Aplicar métodos de compressão direcionados (Magnitude Pruning e Clustering) e aleatórios (Random Pruning) na arquitetura de melhor desempenho;
- Quantificar o impacto das técnicas de poda na redução do custo computacional, medindo o tamanho de armazenamento (em KB) e o tempo de inferência (em milissegundos);
- Analisar a variação da acurácia preditiva (através das métricas MSE, sMAPE e DTW) entre o modelo denso original e as versões submetidas à compressão.

Este documento está estruturado da seguinte forma: o Capítulo 2 apresenta a revisão bibliográfica sobre os conceitos do trabalho. O Capítulo 3 descreve a metodologia e as arquiteturas selecionadas. O Capítulo 4 detalha os experimentos, apresentando e discutindo os resultados da predição e do pruning. Por fim, o Capítulo 5 expõe as conclusões, as limitações encontradas e os possíveis direcionamentos para trabalhos futuros.

2 TRABALHOS RELACIONADOS

2.1 O PAPEL DO APRENDIZADO DE MÁQUINA NO SETOR ENERGÉTICO

O avanço acelerado das tecnologias de computação cria um desafio significativo, visto que o elevado consumo energético dos dispositivos resulta em um considerável impacto ambiental atrelado às emissões de carbono (Henderson et al., 2020; Anthony; Kanding; Selvan, 2020). Para que a Inteligência Artificial (IA) atue como uma ferramenta eficaz na mitigação das mudanças climáticas, é necessário que seus algoritmos tenham seu gasto energético mensurado e otimizado, evitando comportamentos computacionais que não demonstrem as emissões reais do processamento.

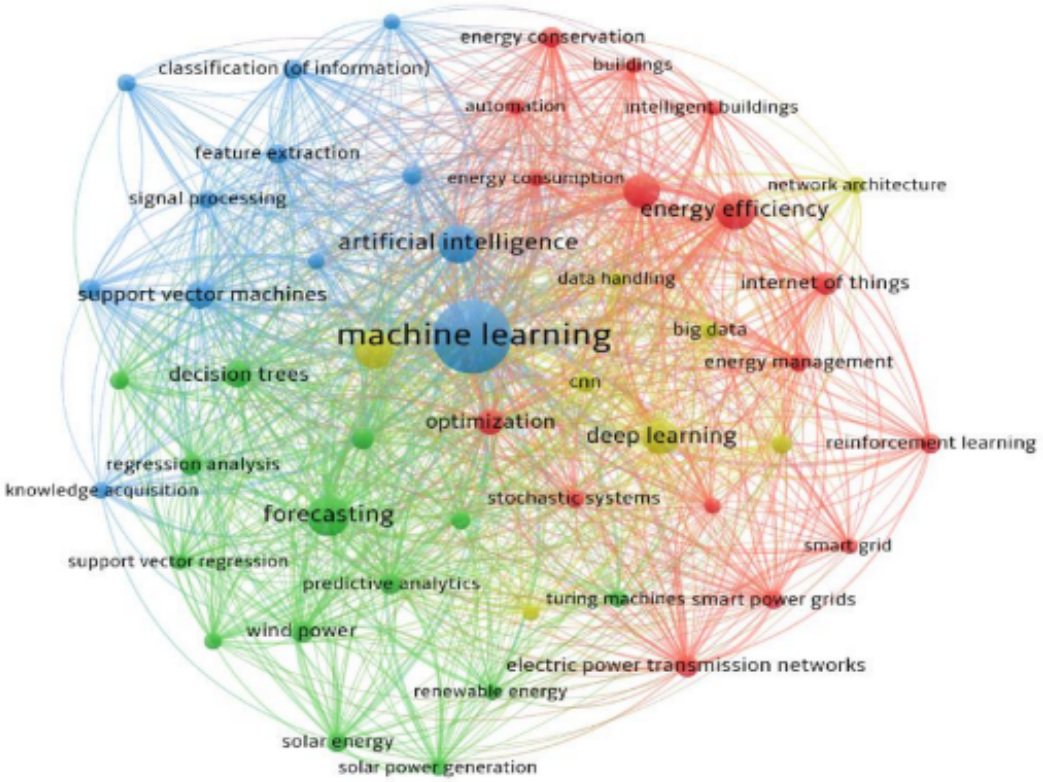
Nesse contexto, o Aprendizado de Máquina (ML — do inglês, Machine Learning) surge como uma abordagem promissora, tanto para a otimização quanto para a predição. O uso de fontes de energia renovável, com ênfase na energia solar, nas redes elétricas tradicionais constitui um grande desafio para o setor energético. Historicamente, a estimativa da geração de energia dependia de modelos matemáticos clássicos. Contudo, essas abordagens encontram severas limitações ao tentar mapear relações complexas e não lineares, características das variáveis climáticas. Com o avanço tecnológico, esses métodos vêm sendo substituídos por modelos de ML (Voyant et al., 2017; Barhmi et al., 2024). Ao analisar padrões históricos e prever demandas de forma dinâmica, esses algoritmos viabilizam uma operação inteligente, antecipando flutuações na geração e facilitando o balanceamento ótimo entre a oferta e a demanda de energia.

2.2 TRABALHOS RECENTES QUE DISCUTEM O CONSUMO ENERGÉTICO/ENERGIA COM APRENDIZADO DE MÁQUINA

A gestão eficiente de energia enfrenta o desafio crítico de conciliar o equilíbrio entre oferta e demanda com a minimização do impacto ambiental. A dependência contínua de combustíveis fósseis exige atenção redobrada a essas questões estruturais (Shivam; Tzou; Wu, 2021; Somu; MR; Ramamritham, 2021). Considerando que a geração vinda de matrizes renováveis é baseada em fatores climáticos, o desenvolvimento de métodos preditivos torna-se essencial para antecipar flutuações e gerenciar o consumo de forma otimizada (Foley et al., 2012; Musbah; Aly; Little, 2021). A Figura 1 ilustra a grande ligação entre o setor energético e o Aprendizado de Máquina .

Além da previsão direta na rede elétrica, os modelos de ML viabilizam inovações estruturais em outras frentes. Trabalhos recentes exploram, por exemplo, o uso de Redes Neurais Artificiais (ANN) para otimizar os processos de fabricação de baterias de íon-lítio , uma etapa vital para garantir a estabilidade e a viabilidade do armazenamento de energia (Turetskyy et al., 2021).

Figura 1 – Mapa de coocorrência de palavras-chave na literatura sobre Aprendizado de Máquina e Energia.



Fonte: Forootan et al. (2022)

2.3 ESTRATÉGIAS PARA REDUÇÃO DO CONSUMO ENERGÉTICO DO APRENDIZADO DE MÁQUINA

Historicamente, a comunidade de IA priorizou precisão ao invés da eficiência computacional, fenômeno impulsionado em grande parte pela escassez de modelos de medição de energia nos frameworks populares de desenvolvimento (Abadi et al., 2016). Para reverter esse paradigma, surgiram ferramentas de software dedicadas a medir o consumo energético e as emissões de carbono associadas, além do estabelecimento de métricas e benchmarks que correlacionam o ganho de precisão ao orçamento computacional exigido (Henderson et al., 2020; Anthony; Kanding; Selvan, 2020; Reddi et al., 2020).

Uma das principais estratégias para a redução do consumo envolve a análise do custo-benefício ao longo de todas as fases do ciclo de vida do modelo. O custo computacional amortizado deve ser avaliado sob uma perspectiva abrangente, possibilitando decisões arquiteturais estratégicas — como o aumento intencional da carga computacional durante o treinamento com o objetivo de reduzir significativamente os custos na fase de inferência (Bommasani et al., 2021).

No caso dos algoritmos, a literatura destaca métodos diretos para incrementar a eficiência das redes neurais. As principais abordagens englobam a compressão de modelos, a criação de técnicas

que demandam menos dados para aprendizado e a redução da frequência de re-treinamento (Hinton; Vinyals; Dean, 2015; Cai et al., 2019; Pfeiffer et al., 2020). A implementação efetiva dessas estratégias exige que os pesquisadores foquem no equilíbrio entre precisão e gasto energético, questionando criticamente se grandes ganhos de desempenho justificam aumentos substanciais no custo computacional (Schwartz et al., 2020; He et al., 2016; Albanie, 2016).

2.4 REDES NEURAIS RECORRENTES

No processamento de dados sequenciais ou séries temporais, as redes neurais tradicionais apresentam deficiências notáveis. Para contornar essa limitação, as Redes Neurais Recorrentes (RNNs - Recurrent Neural Networks) demonstram superioridade na captura de relações de dependência ao longo de uma sequência de dados (Yu et al., 2019; Alhumoud; Wazrah, 2022; Beltozar-Clemente et al., 2024).

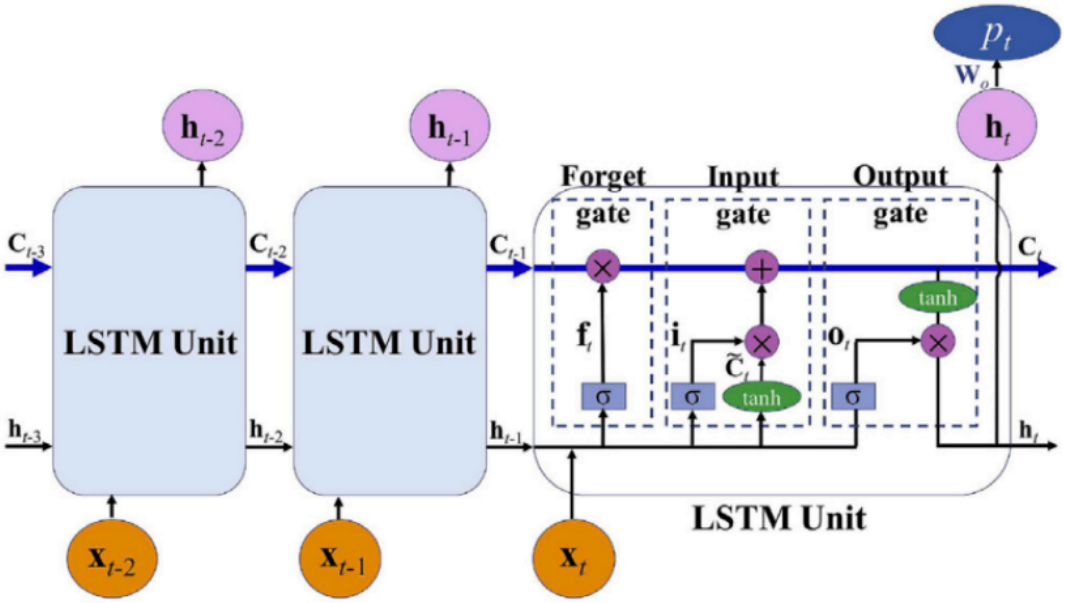
Enquanto novas arquiteturas focam primeiro no paralelismo, as RNNs focam em tarefas que possuem uma estrutura temporal explícita (Zhao et al., 2024). Tais redes preservam informações históricas por meio de conexões retroalimentadas associadas a atrasos de tempo, o que as capacita a detectar correlações temporais distantes (Elman, 1990; Rumelhart; Hinton; Williams, 1986; Werbos, 1988).

Embora o propósito fundamental das RNNs seja o aprendizado de dependências de longo prazo, o processamento de informações por intervalos de tempo muito extensos impõe um severo desafio computacional (Bengio; Simard; Frasconi, 1994). A resolução desse problema envolveu a introdução de mecanismos de memória explícita, resultando na criação da arquitetura LSTM (Long Short-Term Memory) (Hochreiter; Schmidhuber, 1997). As LSTMs provaram-se particularmente robustas ao mitigar a dificuldade de retenção de informações históricas em séries temporais complexas (Zhao et al., 2024). Posteriormente, buscando uma eficiência maior, a arquitetura GRU (Gated Recurrent Unit) foi proposta para capturar, de forma adaptativa, às relações temporais (Cho et al., 2014).

2.4.1 LSTM

A arquitetura LSTM opera basicamente com dois vetores de estado: a memória de longo prazo (representada pelo estado da célula) e o estado de curto prazo (representado pelas ativações transitórias ou estado oculto) (DiPietro; Hager, 2020). A base dessa topologia fica em seu mecanismo de controle de fluxo de informação, composto por portas (gates), conforme esquematizado na Figura 2: a porta de esquecimento (forget gate) determina quais informações devem ser descartadas da célula; a porta de entrada (input gate) controla a adição de novos dados; e a porta de saída (output gate) regula o estado oculto que será repassado adiante (Hochreiter; Schmidhuber, 1997; Muneer et al., 2022). O estado de curto prazo atua como a saída imediata da célula, ao passo que o estado da célula percorre a rede sofrendo atualizações lineares reguladas por essas portas (Al-Selwi et al., 2023).

Figura 2 – Arquitetura interna de uma célula LSTM e suas respectivas portas de controle.



Fonte: Wei et al. (2021)

As redes LSTM foram concebidas primeiramente para contornar a incapacidade das RNNs tradicionais em aprender dependências longas, um problema matematicamente originado pelo desaparecimento (vanishing) e explosão (exploding) de gradientes (Sen; Mehtab, 2022; Hochreiter; Schmidhuber, 1996). Para evitar tais falhas, a arquitetura incorporou o conceito de um carrossel de erro constante (CEC) no núcleo de suas células (Hochreiter; Schmidhuber, 1997).

Matematicamente, a dinâmica da célula LSTM em um instante de tempo t é governada pelas seguintes equações, propostas em sua formulação original (Hochreiter; Schmidhuber, 1997) e consolidadas na literatura clássica. Onde σ representa a função de ativação sigmoide, W e b são as matrizes de peso e vetores de viés, e $\odot$ denota a multiplicação elemento a elemento (Hadamard product):

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

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \tag{2}$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \tag{3}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{4}$$

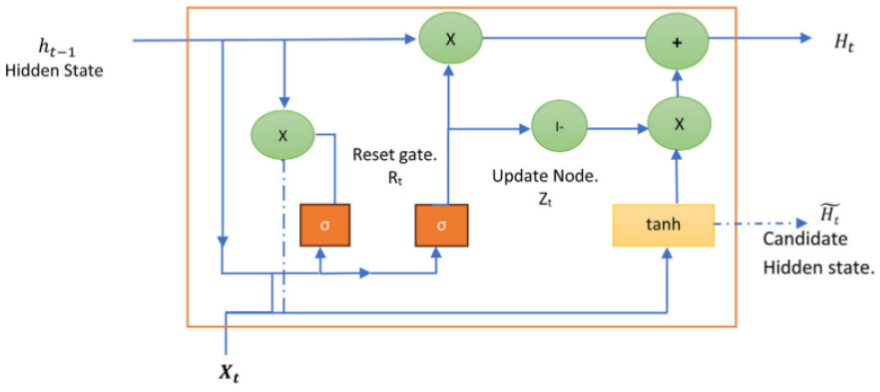
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \tag{5}$$

$$h_t = o_t \odot \tanh(C_t) \tag{6}$$

2.4.2 GRU

Motivados pelo alto custo computacional e elevado número de parâmetros característicos das LSTMs, Cho et al. propuseram, em 2014, a arquitetura GRU (Cho et al., 2014). A GRU apresenta um design estruturalmente mais simplificado (Abbaspour et al., 2020). O objetivo principal dessa arquitetura é oferecer uma topologia mais leve que resolva o problema dos gradientes evanescentes, mantendo, simultaneamente, um desempenho preditivo comparável na retenção de dependências de longo prazo (Cho et al., 2014). O funcionamento de uma unidade GRU baseia-se em duas estruturas, ilustradas na Figura 3: a porta de redefinição (reset gate) e a porta de atualização (update gate) (Dutta; Kumar; Basu, 2020).

Figura 3 – Estrutura simplificada de uma Gated Recurrent Unit (GRU).



Fonte: Waqas & Humphries (2024)

Diferente da LSTM, a GRU elimina a necessidade de um estado de célula isolado e junta as funções das portas de entrada e de esquecimento em uma única porta de atualização conjunta, o que reduz muito o volume de operações matemáticas necessárias (Abbaspour et al., 2020). Nesse arranjo, a porta de redefinição determina matematicamente o que deve ser esquecido, enquanto a porta de atualização calcula o quanto da informação histórica deve ser propagada para o instante seguinte (Dutta; Kumar; Basu, 2020).

A simplificação computacional proposta pela GRU (Cho et al., 2014) pode ser observada através do seu sistema de equações fundamentais (Equações 7 a 10), que utiliza apenas os estados ocultos sem uma memória de célula externa:

$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t] + b_z) \tag{7}$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t] + b_r) \tag{8}$$

$$\tilde{h}_t = \tanh(W_h \cdot [r_t \odot h_{t-1}, x_t] + b_h) \tag{9}$$

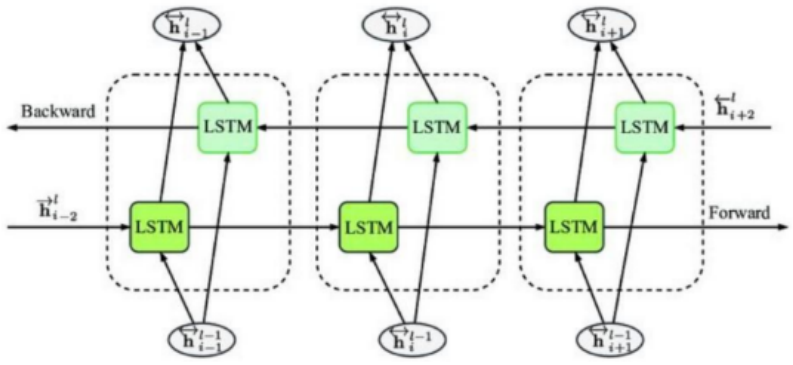
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{10}$$

2.4.3 Redes bidirecionais

Nas arquiteturas tradicionais de RNNs e LSTMs, a saída de um determinado estado depende exclusivamente do histórico de informações passadas (Schuster; Paliwal, 1997). Para superar essa limitação e permitir a consideração do contexto futuro, Schuster e Paliwal introduziram as Redes Neurais Recorrentes Bidirecionais (BRNNs), conceito este que foi estendido para a arquitetura Bi-LSTM (Schuster; Paliwal, 1997; Aldhyani; Alkahtani, 2021; Shiri et al., 2024).

Uma rede Bi-LSTM é estruturada por duas camadas paralelas que processam a sequência em direções temporais opostas, como demonstrado na Figura 4: uma camada analisa os dados no sentido progressivo (da esquerda para a direita) e a outra no sentido regressivo (da direita para a esquerda) (Liciotti et al., 2020). As representações geradas por cada uma dessas camadas são combinadas matematicamente (por exemplo, via concatenação ou soma ponderada) para compor a saída da rede naquele instante (Liciotti et al., 2020).

Figura 4 – Esquema de processamento temporal em uma rede recorrente bidirecional.



Fonte: Aldhyani & Alkahtani (2021)

A principal vantagem do processamento bidirecional é a capacidade de capturar a influência de ambos os contextos temporais sobre um evento (Aldhyani; Alkahtani, 2021; Shiri et al., 2024). Ao integrar a totalidade das informações em ambas as direções, esses modelos conseguem mapear as dependências temporais com precisão superior (Shiri et al., 2023).

2.5 PRUNING

No campo da compressão de redes neurais, a técnica de poda (pruning) tem sido historicamente aplicada com maior ênfase na resolução de problemas de classificação (Li; Yu; Zhou, 2012; Tsoumakas; Partalas; Vlahavas, 2009; Caruana et al., 2004). Em contrapartida, observa-se na literatura uma escassez de trabalhos que investiguem a aplicação destas estratégias de compressão na previsão de séries temporais dinâmicas (Saadallah; Priebe; Morik, 2019; Krikunov; Kovalchuk, 2015).

Dentre os métodos de poda existentes, as abordagens baseadas em similaridade buscam identificar parâmetros duplicados ou conexões que operem de maneira redundante na rede (Han et al., 2016; Zhou et al., 2018b; Li; Zhu; Sun, 2019). Já os métodos baseados em magnitude avaliam a relevância das conexões a partir de métricas locais, como o valor absoluto dos pesos (Weigend; Rumelhart; Huberman, 1991; Zhou et al., 2018a). Outra estratégia engloba os métodos de sensibilidade, cujo princípio consiste em estimar o impacto que a remoção de determinados pesos causaria na função de perda (loss) do modelo (LeCun; Denker; Solla, 1989; Lee; Ajanthan; Torr, 2018).

Além da poda, outros métodos complementares de otimização são a destilação de conhecimento (knowledge distillation) (Hinton; Vinyals; Dean, 2015), a fatoração de matrizes de baixo posto (low-rank factorization) (Jaderberg; Vedaldi; Zisserman, 2014; Lin et al., 2018) e a quantização numérica de parâmetros (Courbariaux et al., 2016; Jacob et al., 2018).

Formalmente, conforme descrito na literatura de compressão de redes (Vadera; Ameen, 2022), a técnica de *Magnitude Pruning* atua aplicando uma máscara binária $M \in \{0, 1\}$ sobre a matriz de pesos W da rede. Um peso $w_{i,j}$ é mantido apenas se o seu valor absoluto for superior a um limiar estabelecido τ, conforme a Equação 11:

$$m_{i,j} = \begin{cases} 1, & \text{se } |w_{i,j}| \geq \tau \\ 0, & \text{se } |w_{i,j}| < \tau \end{cases} \tag{11}$$

Desta forma, o novo peso efetivo da rede torna-se o produto do peso original pela sua respectiva máscara (Equação 12), resultando matematicamente na esparsidade da arquitetura:

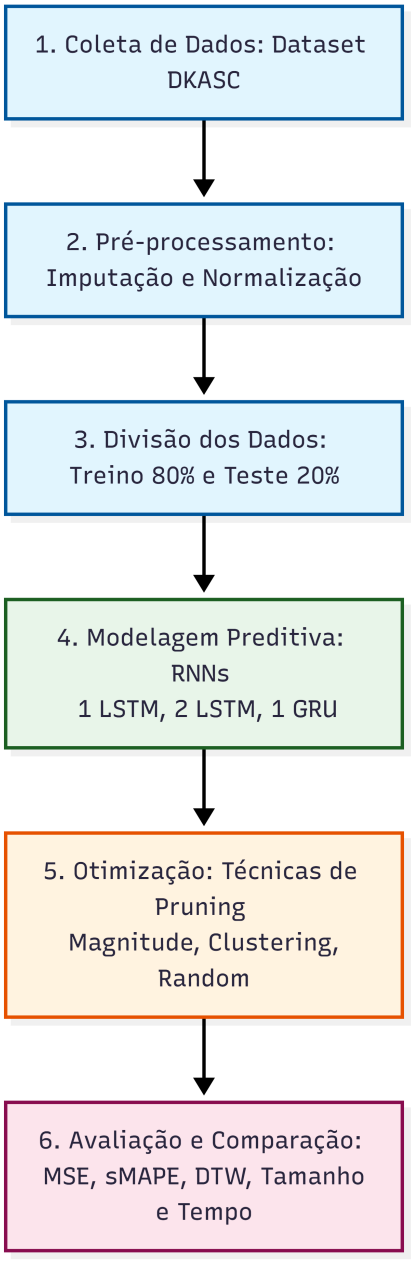
$$w'_{i,j} = w_{i,j} \times m_{i,j} \tag{12}$$

3 METODOLOGIA PROPOSTA

A Metodologia selecionada para esse trabalho foi utilizar Redes Neurais Recorrentes (RNNs) para a predição de séries temporais de energia solar e, posteriormente, aplicar Pruning para otimizar a eficiência energética e computacional dos modelos

Para melhor visualização e compreensão das etapas desenvolvidas neste trabalho, a Figura 5 ilustra o pipeline metodológico completo, desde a coleta e tratamento inicial dos dados do DKASC até a avaliação final dos modelos otimizados por *pruning*.

Figura 5 – Pipeline Metodológico do Trabalho.



Fonte: Elaborada pelo autor.

3.1 PREDIÇÃO

- **Fundamentação e Topologias:** O trabalho utilizou Redes Neurais Recorrentes (RNNs) devido à sua capacidade intrínseca de processar sequências temporais. A definição das topologias testadas (uma e duas camadas LSTM e uma camada GRU) foi estabelecida de maneira empírica, visando avaliar a correlação entre a complexidade estrutural e a eficácia preditiva. Destaca-se um aprimoramento metodológico nesta etapa em relação aos ensaios preliminares: enquanto os testes iniciais basearam-se em execuções únicas (*one-shot*), o protocolo experimental atual estabeleceu maior rigor estatístico. Devido à natureza estocástica da inicialização de pesos em redes neurais, todas as métricas de predição, erro e desempenho computacional reportadas correspondem à média aritmética de 5 execuções independentes, acompanhadas de seus respectivos desvios-padrão. Esta abordagem mitiga o viés de inicialização, garantindo maior robustez e confiabilidade na comparação entre os modelos densos e os submetidos à poda.

- **Arquitetura:** Todas as arquiteturas são seguidas por 10 camadas densas com as seguintes unidades de neurônios: 64,64,32,16,32,8,8,4,2, e uma camada final de saída com 1 neurônio.

- **Parâmetros da Série Temporal:** A série temporal utilizada possui a frequência de amostragem original do conjunto de dados do DKASC (registros a cada 5 minutos). No que tange à formatação dos dados para o modelo preditivo, optou-se por uma abordagem focada nos atributos do próprio instante. A janela de entrada da rede é composta por 15 variáveis preditoras referentes ao instante atual (t), remodeladas sequencialmente, visando prever a variável alvo correspondente a esse mesmo horizonte (horizonte imediato).

- **Treinamento:** Os modelos são treinados por 100 épocas (*epochs*), utilizando um tamanho de *batch* de 32. O algoritmo Adam é usado como otimizador, e a função de perda (*loss function*) é o *Mean Squared Error* (MSE). Optou-se por não utilizar métodos de parada antecipada (*Early Stopping*). Dessa forma, garantiu-se que as redes completassem o ciclo integral de épocas definido para o experimento, permitindo a observação do decaimento da função de perda.

3.2 PRUNING

- **Estratégia de Otimização:** O *pruning* foi aplicado na arquitetura de melhor desempenho identificada na fase de predição preliminar. Este modelo denso original (GRU com 64 unidades) será adotado daqui em diante como o modelo de referência (*Baseline*). Isso garante que a otimização de eficiência seja comparada diretamente com a melhor performance de precisão já alcançada pelo algoritmo intacto.

- **Técnicas de Poda:** A aplicação de pruning será guiada pelo trabalho de Vadera & Ameen (2022) , que descreve métodos como o magnitude-based pruning e o sensitivity analysis.
- **Otimização de Eficiência:** O objetivo é reduzir a complexidade e a dimensão das redes , avaliando o ganho de eficiência energética dos modelos podados através de métricas de FLOPS e outros indicadores de desempenho (Xu et al., 2021).

4 EXPERIMENTOS

Os experimentos foram divididos em três etapas: definição do ambiente de execução (Setup), caracterização completa do conjunto de dados (Dataset) e apresentação dos resultados preliminares dos modelos de predição.

4.1 SETUP (CONFIGURAÇÃO EXPERIMENTAL)

O ambiente computacional e as configurações de software são importantes para garantir a replicabilidade dos resultados.

- **Ambiente e Hardware:** Os modelos foram implementados e treinados no ambiente de desenvolvimento baseado em nuvem Google Colab, utilizando uma GPU NVIDIA Tesla T4 como acelerador gráfico.
- **Bibliotecas e Ferramentas:** A base do desenvolvimento foi a linguagem Python. As redes neurais recorrentes (LSTM e GRU) foram implementadas utilizando a biblioteca Keras do TensorFlow. Para manipulação e pré-processamento de dados, foram utilizadas as bibliotecas Pandas e NumPy, enquanto o Scikit-learn foi essencial para a separação dos conjuntos de dados e técnicas de escalonamento. O cálculo da métrica de similaridade de séries temporais foi realizado com a biblioteca tslearn.
- **Medição de Eficiência Energética:** O foco na otimização da eficiência será quantificado através de duas abordagens:
    1. Avaliação da complexidade utilizando o indicador FLOPS (operações de ponto flutuante por segundo) e a contagem de parâmetros, conforme as métricas de otimização de Deep Learning citadas por Xu et al. (2021)
    2. Medição direta do tempo de treinamento por época e do tempo de inferência nos modelos otimizados, demonstrando o ganho de velocidade.

4.2 DATASET (CONJUNTO DE DADOS)

O dataset escolhido para este trabalho o Desert Knowledge Australia Solar Centre (DKASC).

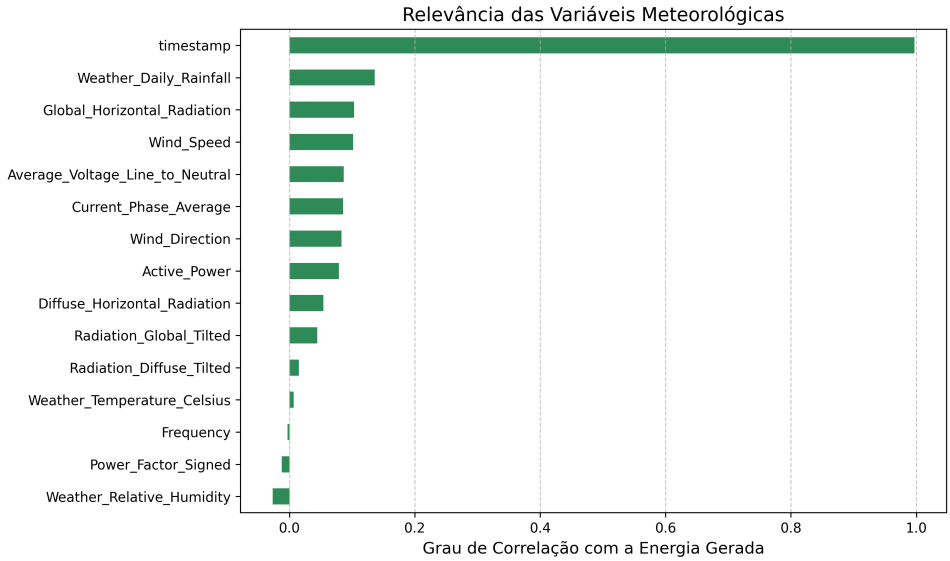
- **Tamanho e Separação:** O conjunto de dados bruto foi processado, resultando em um total de 724.261 registros. Para o treinamento e avaliação dos modelos, o conjunto de dados foi dividido em dois subconjuntos:
    - **Treinamento:** 80% dos dados (aproximadamente 579.409 amostras).
    - **Teste:** 20% dos dados.

- **Features:** Dentre as 15 variáveis selecionadas, a variável alvo (*target*) definida para o treinamento dos modelos foi a **Energia Ativa Acumulada (kWh)**. Esta escolha permite que o modelo aprenda a progressão temporal da energia total gerada pela usina, integrando as variações meteorológicas em um índice de produtividade acumulada.

  Para fundamentar a utilização em conjunto destas variáveis meteorológicas (como irradiância e temperatura) e características elétricas locais, realizou-se uma análise de importância das *features* (*Feature Importance*). O método escolhido foi o Coeficiente de Correlação de Pearson, que quantifica a força e a direção da dependência linear entre cada variável sensoriada e a variável alvo (Montgomery; Runger, 2014). Conforme ilustrado naFigura 6, este procedimento comprovou que as variáveis selecionadas possuem relevância estatística e impacto direto na capacidade preditiva do modelo, justificando a sua manutenção e descartando a necessidade de exclusão por falta de correlação.

  A incorporação desse contexto ambiental amplo visa permitir que o modelo aprenda a relação termodinâmica específica daquela instalação. Embora essa abordagem torne os pesos da rede originalmente dependentes das características técnicas da usina DKASC, a literatura indica que a topologia da rede e a técnica de compressão aplicadas podem ser facilmente generalizadas para outras usinas através de *Transfer Learning* (Transferência de Aprendizado) (Chu et al., 2024), necessitando apenas de um ajuste fino (*fine-tuning*) com uma pequena amostra de dados do novo local para recalibrar o modelo.

Figura 6 – Importância das Variáveis Meteorológicas e Elétricas via Correlação de Pearson.



Fonte: Elaborada pelo autor.

- **Pré-processamento:** Para tratar valores omissos, foi empregada a imputação por média (SimpleImputer). Devido à presença de outliers e à variabilidade extrema inerente aos dados climáticos e de energia, foi aplicado o RobustScaler para a reescala das features de entrada, tornando a distribuição dos dados mais robusta ao treinamento da rede.

- **Distribuição (PDF):** A distribuição da variável alvo (Energia Ativa) é característica de dados solares, apresentando uma natureza bimodal: um pico acentuado em zero (correspondente ao período noturno e sem geração) e um segundo pico durante o dia, refletindo a potência de geração mais comum.

4.3 MÉTRICAS DE AVALIAÇÃO

Para quantificar e avaliar o desempenho dos modelos preditivos, foram adotadas três métricas principais: o Erro Quadrático Médio (MSE), o Erro Percentual Absoluto Médio Simétrico (sMAPE) e a Distorção Temporal Dinâmica (DTW).

O MSE penaliza de forma mais severa os grandes desvios entre o valor previsto e o valor real, uma vez que as diferenças são elevadas ao quadrado. Ele é expresso matematicamente pela Equação 13:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2 \tag{13}$$

onde n representa o número total de amostras, Y_i é o valor real medido e $\hat{Y}_i$ é o valor predito pelo modelo.

O sMAPE quantifica o erro de previsão de maneira percentual e simétrica. Esta métrica é particularmente útil para séries temporais que possuem valores próximos a zero (como a geração de energia solar durante a noite), pois mitiga o problema de divisão por zero ou valores infinitos encontrado no MAPE tradicional, conforme a Equação 14:

$$sMAPE = \frac{100\%}{n}\sum_{i=1}^{n}\frac{|Y_i - \hat{Y}_i|}{(|Y_i| + |\hat{Y}_i|)/2} \tag{14}$$

Por fim, a métrica de Distorção Temporal Dinâmica (*Dynamic Time Warping* - DTW) foi adotada para avaliar a similaridade topológica entre as séries temporais. Diferente do MSE ou sMAPE, que calculam o erro comparando os dados estritamente ponto a ponto no mesmo instante de tempo (alinhamento linear), o algoritmo DTW encontra o alinhamento ótimo não-linear entre duas sequências. Essa característica é fundamental no contexto fotovoltaico, pois permite avaliar se o modelo preditivo foi capaz de aprender o "formato"da curva de geração (os vales noturnos e os picos diurnos), penalizando modelos que apenas traçam médias estáticas.

Matematicamente, dadas duas séries temporais de tamanho n, sendo Y a série real e $\hat{Y}$ a série predita, o algoritmo constrói uma matriz de custo onde cada coordenada (i, j) armazena a distância (neste caso, Euclidiana) entre o ponto Y_i e o ponto $\hat{Y}_j$. O cálculo do custo acumulado mínimo $D(i, j)$ para se encontrar o caminho de distorção ótimo (*warping path*) é definido

recursivamente pela Equação 15:

$$D(i,j) = |Y_i - \hat{Y}_j| + \min \begin{cases} D(i-1,j) \\ D(i,j-1) \\ D(i-1,j-1) \end{cases} \tag{15}$$

A distância final DTW reportada nos resultados corresponde ao valor acumulado na última célula da matriz de custo, ou seja, $D(n,n)$. Quanto menor este valor, maior é a fidelidade do modelo em acompanhar a dinâmica temporal do fenômeno físico.

4.4 RESULTADOS DOS EXPERIMENTOS DE PREDIÇÃO

Para realizar as predições, foram exploradas três variações de arquiteturas de redes neurais recorrentes (RNNs), cada uma configurada com 10 camadas ocultas. As arquiteturas específicas incluíam uma com uma única camada de Long Short-Term Memory (LSTM), outra com duas camadas de LSTM e uma terceira com uma camada de Gated Recurrent Unit (GRU).

Detalhando a arquitetura, as configurações implementadas para cada variação consistiram em suas respectivas camadas recorrentes seguidas por um bloco de 10 camadas ocultas densas totalmente conectadas (fully connected layers), distribuídas da seguinte forma:

- **Uma camada de LSTM:** Camada recorrente seguida por camadas densas com 64, 64, 32, 16, 32, 8, 8, 4, 2 e 1 neurônio(s).
- **Duas camadas de LSTM:** Duas camadas recorrentes em série, seguidas pelo mesmo bloco de 10 camadas densas (64, 64, 32, 16, 32, 8, 8, 4, 2 e 1).
- **Uma camada de GRU:** Camada recorrente seguida pelas 10 camadas densas (64, 64, 32, 16, 32, 8, 8, 4, 2 e 1 neurônios).

Os resultados da predição demonstram a eficácia das Redes Neurais Recorrentes na modelagem da geração fotovoltaica, concordando com a literatura apresentada no Capítulo 2. Ao analisar as métricas médias de erro em 5 execuções, consolidadas na Tabela 1, nota-se que a arquitetura GRU com 64 unidades ocultas obteve o menor sMAPE médio (7,58%), demonstrando clara vantagem na minimização do erro percentual absoluto em comparação aos modelos LSTM, que não conseguiram convergir para a curva da série temporal sob a mesma topologia densa herdada da Iniciação Científica. Considerando este desempenho superior e o fato de possuir uma arquitetura nativamente mais simplificada, a GRU foi selecionada como a topologia ideal para ser submetida aos experimentos de otimização por poda (pruning) na etapa seguinte.

Tabela 1 – Valores das métricas médias de cada arquitetura e do modelo estatístico na predição.

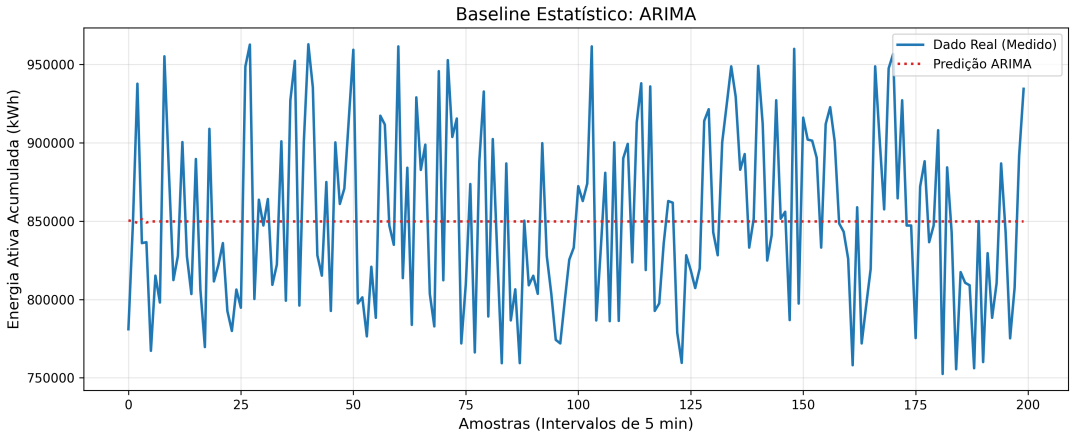
Métrica	**ARIMA (Baseline)**	**1 LSTM**	**2 LSTM**	**GRU (64)**
sMAPE (%)	**6,00**	89.450,13 ± 0,00	90.084,89 ± 0,00	7,58 ± 0,15
DTW (kWh)	595.083.155	336.635.671 ± 0	279.451.749 ± 0	**2.655.337 ± 169.543**
MSE (kWh^2)	3.514.740.193	738.542.298 ± 0	**539.746.851 ± 0**	895.389.658 ± 107.102.951

Fonte: Elaborada pelo Autor.

Para oferecer uma perspectiva visual mais detalhada desses resultados, as Figuras Figura 7, Figura 8, Figura 9, Figura 10 e Figura 11 foram elaboradas representando, respectivamente, o modelo estatístico ARIMA e as arquiteturas de redes neurais propostas. Nesta representação gráfica, as séries temporais reais estão destacadas em azul, enquanto as previstas estão em laranja. Cumpre ressaltar que o modelo ARIMA foi executado utilizando seus hiperparâmetros padrão (default), sem a aplicação de métodos de otimização automatizada (como grid search) para a busca de defasagens ótimas. Consequentemente, a ausência de uma calibração fina das janelas de sazonalidade limitou a capacidade do modelo em capturar a complexa não-linearidade dos dados climáticos, conforme observado na Figura Figura 7.

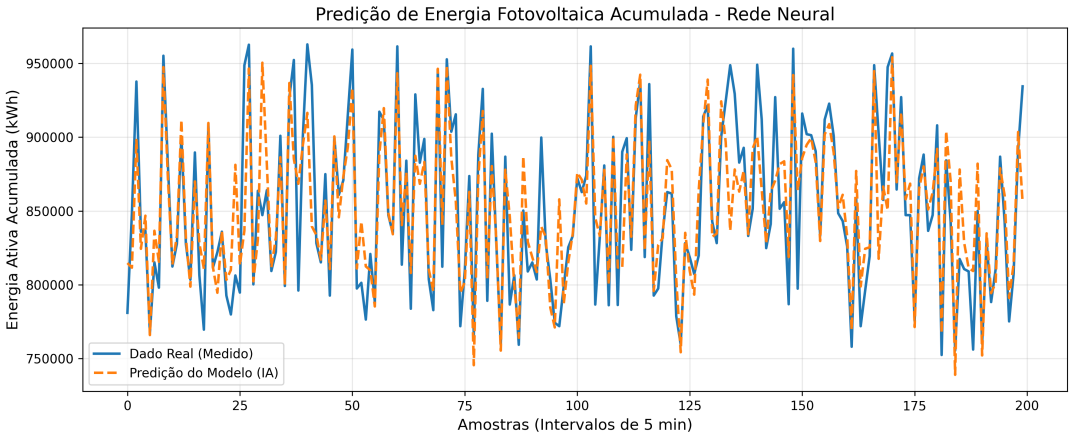
A análise visual possibilita uma melhor compreensão das tendências e discrepâncias, contribuindo para uma avaliação abrangente do desempenho. Para fins de visualização qualitativa, estes gráficos representam o comportamento temporal dos modelos a partir de uma execução representativa (one-shot), enquanto as métricas quantitativas definitivas, considerando a variância das inicializações, encontram-se resumidas na Tabela 1.

Figura 7 – Gráfico do Resultado real e previsto (ARIMA).



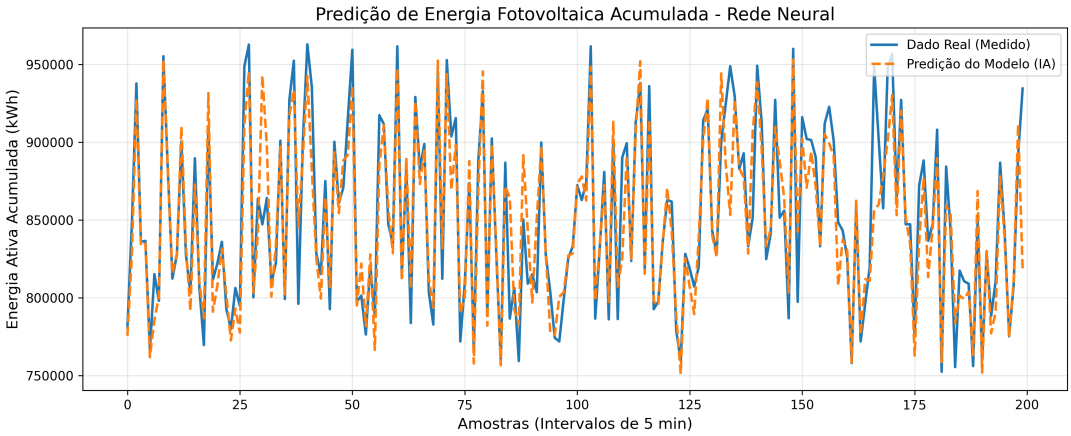
Fonte: Elaborada pelo autor.

Figura 8 – Gráfico do Resultado real e previsto (1 LSTM).



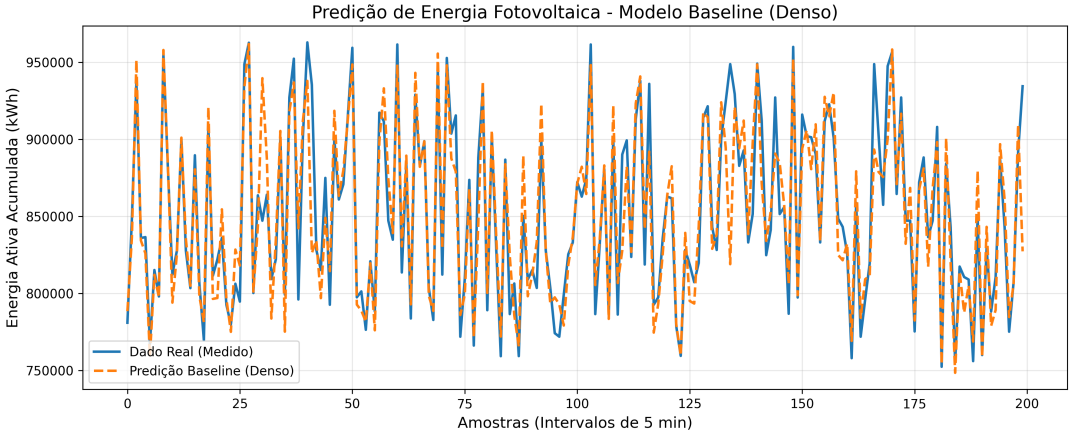
Fonte: Elaborada pelo autor.

Figura 9 – Gráfico do Resultado real e previsto (2 LSTM).



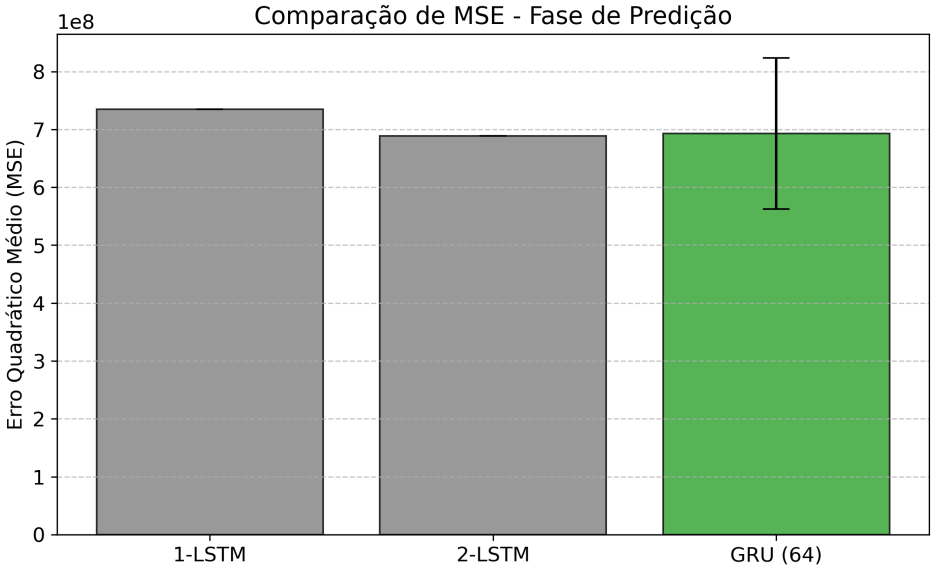
Fonte: Elaborada pelo autor.

Figura 10 – Gráfico do Resultado real e previsto (1 GRU).



Fonte: Elaborada pelo autor.

Figura 11 – Comparação do Erro Quadrático Médio (MSE) entre as arquiteturas na predição.



Fonte: Elaborada pelo autor.

4.5 DISCUSSÃO DOS RESULTADOS DA PREDIÇÃO

Os resultados da predição demonstram dinâmicas distintas entre as abordagens avaliadas. Para estabelecer um referencial de controle, implementou-se o modelo estatístico linear ARIMA. Conforme a Tabela 1, este modelo obteve um sMAPE de 6,00%. No entanto, este indicador isolado não reflete a precisão real da série devido à característica do sMAPE em apresentar valores reduzidos em intervalos com limites próximos a zero (períodos noturnos). A limitação do modelo estatístico em mapear a variabilidade climática é evidenciada pelo elevado erro quadrático (MSE superior a 3,5 bilhões) e pela métrica de distância temporal (DTW de 595 milhões). A Figura 7 ilustra esta limitação: o ARIMA foi incapaz de modelar as oscilações cíclicas de irradiância, convergindo para uma tendência média constante.

Em contrapartida, a rede GRU, apesar do sMAPE de 7,58%, demonstrou capacidade de acompanhar a sazonalidade diária da geração fotovoltaica, registrando um DTW significativamente inferior (2,6 milhões). O erro percentual discrepante observado nas arquiteturas LSTM (aproximadamente 90.000%) decorre de um colapso de representação do modelo que, sob a topologia testada, subestimou a saída em ordens de magnitude próximas ao zero real, gerando instabilidade na métrica percentual.

No campo das Redes Neurais Recorrentes, os modelos LSTM (tanto de uma quanto de duas camadas) apresentaram um fenômeno de colapso de representação. Essa falha se reflete nos valores extremos de sMAPE (na casa de 90.000%) e na ausência de variância nas 5 execuções independentes. Como as LSTMs possuíam apenas 16 unidades em sua camada inicial, acopladas a um extenso bloco de 10 camadas densas lineares herdadas de ensaios preliminares, a rede não teve capacidade dimensional para extrair as características estocásticas do clima, convergindo

também para uma previsão estática. O baixo valor numérico do MSE da 2-LSTM não reflete um aprendizado real da curva, mas apenas a penalização matemática da escala.

A GRU, por sua vez, demonstrou ser a única arquitetura com capacidade de acompanhamento temporal real. Por possuir uma largura de 64 unidades na sua camada recorrente, ela conseguiu evitar o desvanecimento do gradiente e contornar a linearidade das camadas seguintes. Isso é matematicamente comprovado pelo seu excelente DTW de 2,6 milhões (uma redução de distorção absurda quando comparada aos 595 milhões do ARIMA), provando que ela mapeou o formato da curva com precisão. Este desempenho superior e sua arquitetura nativamente mais simplificada justificam a sua seleção como a topologia base para ser submetida aos experimentos de otimização por poda (pruning) na etapa seguinte.

A análise visual dos gráficos revela que as redes apresentaram valores previstos ligeiramente inferiores aos picos reais diurnos, além de estabilizarem estritamente em zero durante o período noturno. A subestimação dos picos de geração é um comportamento característico de modelos otimizados pela função de perda de Erro Quadrático Médio (MSE) (Sharadga; Hajimirza; Balog, 2020). O MSE tende a traçar uma regressão conservadora para evitar penalizações extremas diante de ruídos climáticos isolados (outliers de irradiância). Por outro lado, as previsões zeradas noturnas refletem a eficácia da normalização combinada com as funções de ativação das redes, que corretamente forçaram a saída do modelo a zero na total ausência de radiação solar, respeitando o limite físico do fenômeno.

Sob a perspectiva da sustentabilidade computacional, o treinamento das arquiteturas foi monitorado quanto ao consumo energético. Utilizando uma GPU NVIDIA Tesla T4 ($P = 70$W) durante o tempo total de execução do protocolo experimental de 5 rodadas, incluindo os modelos densos e podados ($t \approx 1,25$h), obteve-se um consumo estimado de $0,0875$ kWh (Equação 16).

$$E = P \times t = 0,07 \text{ kW} \times 1,25 \text{ h} = 0,0875 \text{ kWh} \tag{16}$$

Este valor, fundamentado nos logs de tempo de execução do ambiente de nuvem e nas especificações técnicas do hardware (NVIDIA, 2019), valida a viabilidade do modelo para aplicações reais sem demandar infraestruturas de alto custo energético.

4.6 RESULTADOS DOS EXPERIMENTOS COM PRUNING

A fim de verificar a premissa energética proposta pelo trabalho de mitigar o tempo computacional das redes pré-determinadas, o experimento submeteu o modelo denso à rotina de poda. Na Tabela 2 e na Tabela 3, é possível observar o *benchmark* entre o modelo de *Baseline* contra as simulações baseadas em técnicas de *Magnitude Pruning*, *Clustering* e poda aleatória (*Random Pruning*), sob avaliações exatas de acurácia, memória (em KB) e milissegundos despendidos em inferência.

Tabela 2 – Impacto das técnicas de otimização na acurácia do modelo GRU.

Técnica	**MSE (kWh^2)**	**sMAPE (%)**	**DTW (kWh)**
Baseline (Denso)	**895.389.658 ± 107.102.951**	7,58 ± 0,15	**2.655.337 ± 169.543**
Magnitude Pruning	1.263.532.979 ± 268.813.818	**7,46 ± 0,16**	3.113.971 ± 320.539
Clustering	1.026.194.163 ± 79.342.349	7,49 ± 0,11	2.808.945 ± 130.770
Random Pruning	725.404.601.549 ± 3.518.592.332	199,49 ± 0,97	91.913.078 ± 223.327

Fonte: Elaborada pelo Autor.

Tabela 3 – Avaliação do desempenho computacional e taxas de compressão do modelo GRU.

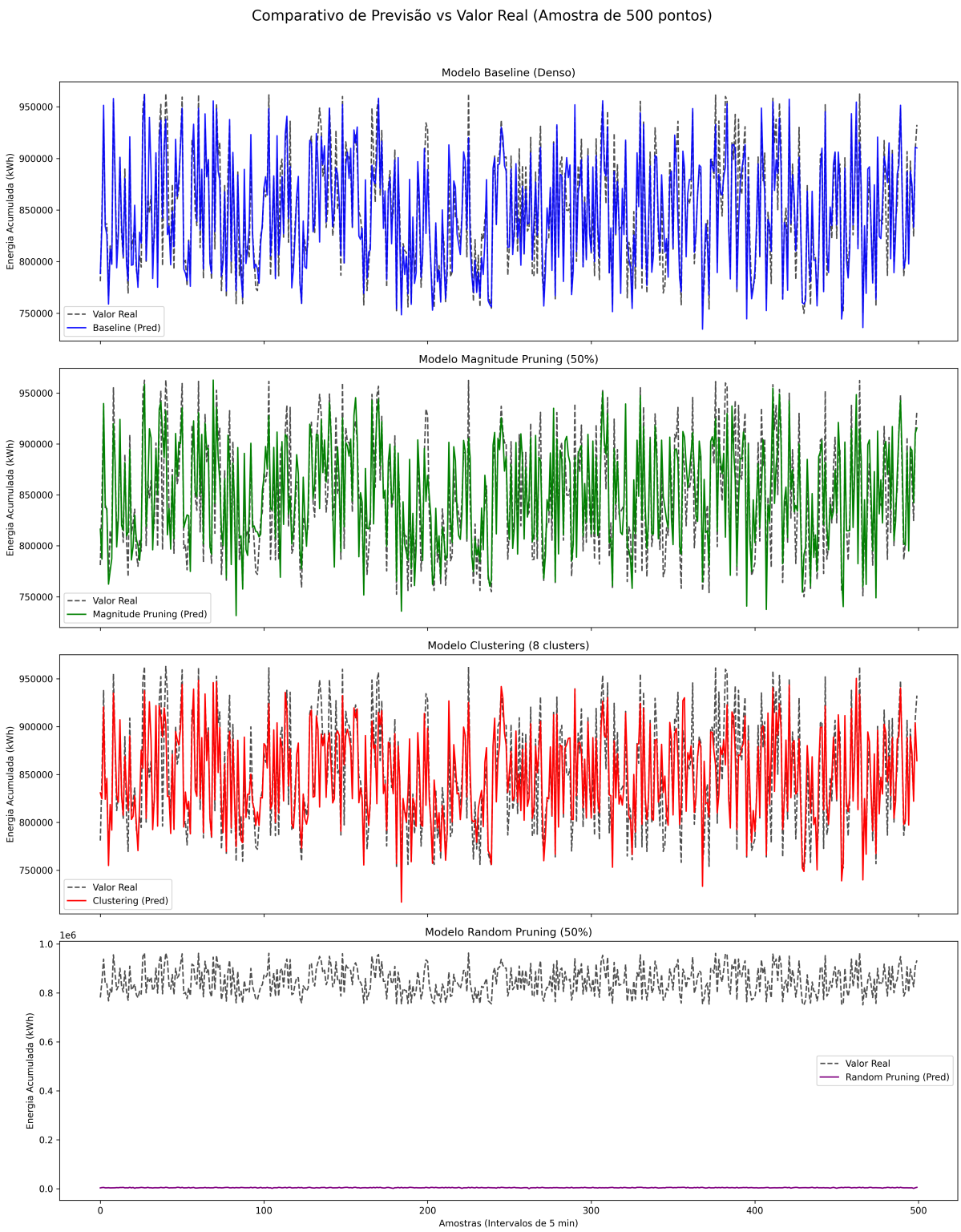
Técnica	**Tamanho (KB)**	**Comp. (KB)**	**Não-Zero**	**Comp. (Pesos)**	**Tempo (ms)**
Baseline (Denso)	353	1,00x	24.721	1,00x	353,43 ± 10,75
Magnitude	**144**	**0,41x**	12.668	0,51x	374,85 ± 46,43
Clustering	**144**	**0,41x**	24.721	1,00x	**330,31 ± 11,71**
Random	353	1,00x	**12.513**	**0,51x**	332,65 ± 22,48

Fonte: Elaborada pelo Autor.

Para melhor visualização do impacto destas técnicas na previsão da série temporal, a Figura 12 ilustra o comparativo entre as predições geradas e os valores reais, considerando uma amostra de 500 pontos. Fica evidente que os modelos otimizados pelas técnicas de *Magnitude Pruning* e *Clustering* mantêm a capacidade de generalização e acompanham a tendência da curva real de maneira quase idêntica ao modelo denso original (Baseline). Por outro lado, a rede submetida a poda aleatória (*Random Pruning*) sofre um colapso e perde completamente sua capacidade preditiva. Para fins de visualização qualitativa do ajuste da curva, este gráfico representa o comportamento temporal do modelo a partir de uma execução representativa (one-shot). As métricas quantitativas definitivas de erro, considerando a variância aleatória, encontram-se resumidas na Tabela 2.

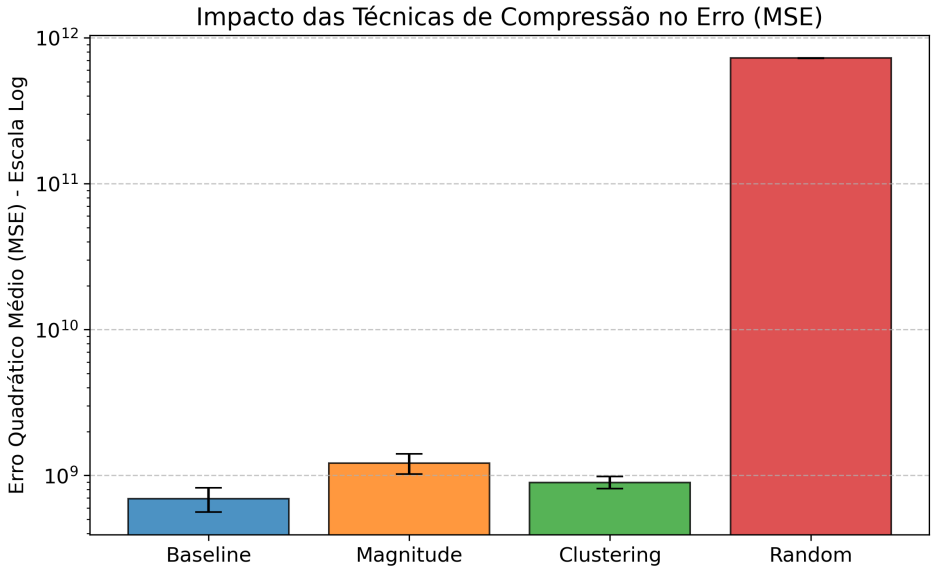
As métricas de erro também corroboram este comportamento. A Figura 13 e a Figura 14 apresentam graficamente como os modelos estruturados por magnitude e clusterização conseguem preservar a integridade das predições, mantendo os níveis de distorção (MSE e sMAPE) em níveis seguros e bastante próximos ao Baseline. Em contrapartida, as figuras revelam como a poda aleatória eleva as falhas absolutas a níveis impraticáveis.

Figura 12 – Comparativo de previsão versus valor real (Amostra de 500 pontos).



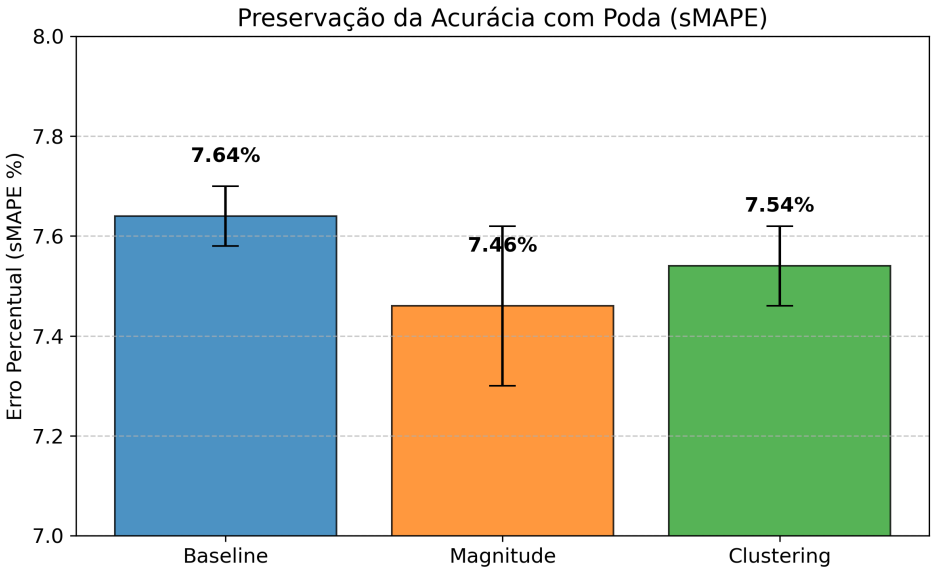
Fonte: Elaborada pelo autor.

Figura 13 – Impacto das técnicas de compressão no Erro Quadrático Médio (escala logarítmica).



Fonte: Elaborada pelo autor.

Figura 14 – Preservação da acurácia do modelo GRU com as técnicas de compressão (sMAPE).

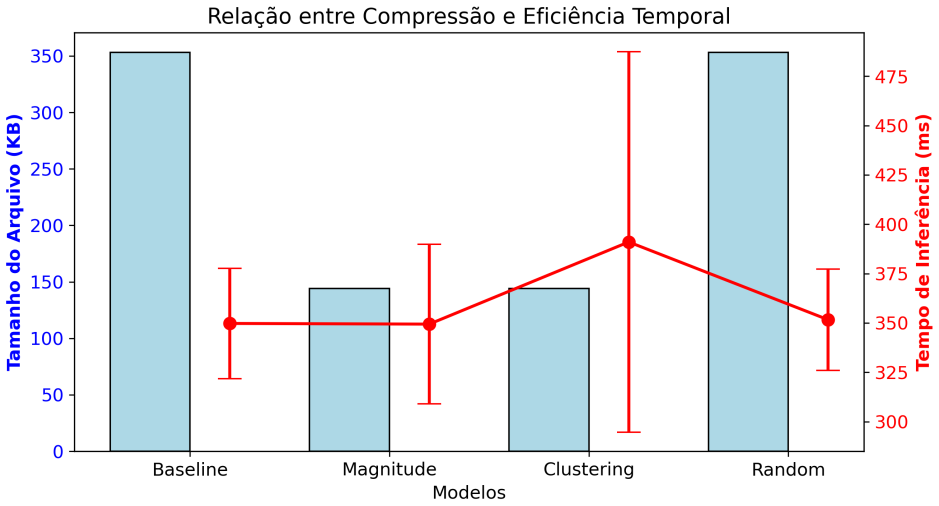


Fonte: Elaborada pelo autor.

Sob a ótica da eficiência computacional, os resultados apresentados na Figura Figura 15 validam as taxas de compressão atingidas pelos métodos aplicados. Conforme discutido na Seção 4.5, o treinamento demanda um investimento energético inicial, que é então otimizado pelas técnicas de poda para garantir maior eficiência durante a inferência. Observa-se que, mantendo-se a acurácia preditiva, as técnicas reduziram o tamanho do modelo em disco de 353 KB para 144 KB, o que representa uma supressão de 59% do seu volume original. Adicionalmente, no caso específico da poda por magnitude, nota-se uma redução de aproximadamente 48,7% na

quantidade de parâmetros não-zero, otimizando a estrutura da rede para dispositivos de baixa capacidade de armazenamento.

Figura 15 – Relação entre a compressão de memória (KB) e a eficiência temporal (ms) do modelo GRU.



Fonte: Elaborada pelo autor.

4.7 DISCUSSÃO DOS RESULTADOS DO PRUNING

A análise dos experimentos de compressão confirma a viabilidade de otimizar redes neurais recorrentes sem prejudicar sua capacidade preditiva. Conforme evidenciado na Tabela 2 e na Figura Figura 14, as técnicas direcionadas — especificamente o Magnitude Pruning e o Clustering — conseguiram preservar a integridade das predições, acompanhando a tendência da curva real de geração de energia de maneira quase idêntica ao modelo denso de referência (*Baseline*).

O modelo submetido ao *Magnitude Pruning* (50%), por exemplo, obteve um sMAPE de 7,46%, superando a precisão da rede original, que registrou 7,58%. Este fenômeno destaca-se como o resultado mais expressivo do experimento, pois evidencia o efeito de *regularização implícita* proporcionado pela poda. Em topologias profundas (como o bloco de 10 camadas densas utilizado), o excesso paramétrico frequentemente induz ao *overfitting* (sobreajuste), situação em que a rede memoriza ruídos específicos dos dados de treinamento em vez de generalizar a dinâmica do sistema fotovoltaico.

Ao forçar o modelo a operar com apenas metade de suas conexões, a técnica de magnitude atuou como um filtro, eliminando os pesos de menor relevância (próximos a zero) que ancoravam a rede a esses ruídos. Consequentemente, a remoção da redundância matemática gerou uma fronteira de decisão mais suave e generalista, permitindo um ganho de acurácia sobre dados não vistos. Em contrapartida, a aplicação do *Random Pruning* cortou conexões fundamentais de maneira cega, resultando em uma falha estrutural severa (elevando o sMAPE a 199,49% e

destruindo a capacidade preditiva). Esse contraste atesta o que é dito na literatura da área: a compressão de redes garante eficiência, mas exige critérios determinísticos para a supressão de pesos.

Sob a ótica da eficiência computacional, conforme sumarizado na Tabela 3, os ganhos quantitativos obtidos validam a aplicação das técnicas de poda. A estratégia de *Magnitude Pruning* resultou em uma taxa de compressão de 59%, reduzindo o tamanho de armazenamento do modelo de 353 KB para 144 KB e suprimindo aproximadamente 48,7% dos parâmetros não-zero (de 24.721 para 12.668). Embora as rotinas de poda demandem maior custo computacional durante a etapa de treinamento para a identificação e mapeamento da relevância dos pesos, este investimento é compensado na fase de inferência. A técnica de *Clustering*, por exemplo, reduziu o tempo de resposta para 330,31 milissegundos, representando um incremento de velocidade em relação ao modelo denso original (353,43 ms).

Esses resultados atendem diretamente à premissa sustentável do trabalho. A diminuição do tempo de inferência e do volume de memória atesta a eficácia da poda para aplicações implementadas na ponta (edge computing) e em tempo real. Ao reduzir a carga computacional associada à tomada de decisão das predições fotovoltaicas, os modelos podados não apenas mantêm a precisão necessária para o balanceamento da rede elétrica, mas também mitigam ativamente o custo computacional do próprio algoritmo. Como apontado na literatura, essa eficiência é um forte indicativo de menor consumo de energia, alinhando a Inteligência Artificial às metas de mitigação do impacto ambiental abordadas na introdução deste projeto.

5 CONCLUSÃO

A mudança energética global em direção a matrizes renováveis, com ênfase na energia solar fotovoltaica, impõe o desafio crítico de gerenciar a irregularidade típica dessas fontes, visando garantir a estabilidade das redes elétricas. Diante dessa complexidade, os algoritmos de Aprendizado de Máquina, especialmente as Redes Neurais Recorrentes (RNNs), mostram-se como ferramentas essenciais para a predição da geração de energia. Contudo, o avanço da Inteligência Artificial trouxe um paradoxo ambiental: a busca contínua por precisão preditiva resultou em modelos com altíssimo custo paramétrico, que o consumo energético na fase de inferência e treinamento pode mascarar os benefícios sustentáveis que buscam promover. Nesse contexto, este trabalho demonstrou sua relevância ao não apenas desenvolver e avaliar arquiteturas preditivas complexas (LSTM e GRU), mas, principalmente, ao propor a otimização dessas redes por meio de técnicas de poda (pruning), visando equilibrar a eficácia da predição com a eficiência computacional e alinhando o desenvolvimento tecnológico aos preceitos da mitigação do impacto climático.

Os resultados obtidos confirmaram a hipótese de que é possível reduzir drasticamente a carga computacional de modelos profundos sem comprometer sua capacidade de generalização em séries temporais complexas. Na etapa de modelagem, constatou-se que a arquitetura GRU de 64 unidades evitou o colapso de representação sofrido pelas LSTMs menos densas, oferecendo a melhor relação de Erro Quadrático Médio (MSE) e Erro Percentual (sMAPE). A principal contribuição deste estudo, contudo, fica na aplicação empírica da compressão. A técnica de Magnitude Pruning reduziu o tamanho da rede em aproximadamente 59% (de 353 KB para 144 KB) e cortou pela metade a quantidade de parâmetros ativos. Surpreendentemente, esse modelo podado manteve uma precisão (sMAPE de 7,46%) ligeiramente superior à da rede densa original (7,58%), possivelmente devido a um efeito de regularização que mitigou o overfitting. Além disso, a técnica de Clustering destacou-se por otimizar o tempo de inferência para 330,31 milissegundos. Em contraste, a falha do Random Pruning validou a necessidade de critérios determinísticos na seleção de pesos.

Apesar dos resultados promissores, é necessário ressaltar as limitações inerentes a este estudo. A avaliação das arquiteturas preditivas e dos métodos de compressão foi conduzida utilizando os dados históricos de uma usina fotovoltaica específica (DKASC). Consequentemente, a generalização do desempenho dessas redes podadas para usinas situadas em diferentes latitudes ou sob regimes climáticos mais instáveis ainda carece de validação empírica. Além disso, o escopo das técnicas limitou-se à poda por magnitude, por agrupamento e aleatória. Por fim, as métricas de tempo de inferência e tamanho de memória foram obtidas em um ambiente de hardware padronizado para pesquisa (nuvem), o que significa que o consumo energético elétrico exato dos algoritmos executados em microcontroladores industriais não foi diretamente medido.

Tendo em vista as limitações apontadas e as descobertas alcançadas, apresentam-se diversas perspectivas para trabalhos futuros. Primeiramente, recomenda-se a expansão das técnicas de otimização, englobando métodos como a quantização de parâmetros (redução da precisão numérica dos pesos de float32 para int8) e a destilação de conhecimento (knowledge distillation), a fim de comparar seus impactos com a poda por magnitude. Outra vertente crucial é a implementação física desses modelos comprimidos em dispositivos embarcados (como FPGAs ou microcontroladores voltados para IoT). Adicionalmente, sugere-se o treinamento e a validação das arquiteturas desenvolvidas utilizando datasets de múltiplas usinas em diferentes regiões do globo, bem como a incorporação de arquiteturas mais recentes, como os modelos baseados em Transformers, avaliando se as estratégias de poda exploradas neste trabalho se traduzem de forma equivalente nessas novas topologias.

REFERÊNCIAS

ABADI, M. et al. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. **arXiv preprint arXiv:1603.04467**, 2016.

ABBASPOUR, S. et al. A comparative analysis of hybrid deep learning models for human activity recognition. **Sensors**, MDPI, v. 20, n. 19, p. 5707, 2020.

AL-SELWI, S. M. et al. Lstm inefficiency in long-term dependencies regression problems. **Journal of Advanced Research in Applied Sciences and Engineering Technology**, v. 30, n. 3, p. 16–31, 2023.

ALBANIE, S. Convnet-burden: estimates of memory consumption and flop counts for various convolutional neural networks. **URL https://github. com/albanie/convnet-burden**, 2016.

ALDHYANI, T. H.; ALKAHTANI, H. A bidirectional long short-term memory model algorithm for predicting covid-19 in gulf countries. **Life**, MDPI, v. 11, n. 11, p. 1118, 2021.

ALHUMOUD, S. O.; WAZRAH, A. A. A. Arabic sentiment analysis using recurrent neural networks: a review. **Artificial Intelligence Review**, Springer, v. 55, n. 1, p. 707–748, 2022.

ANTHONY, L. F. W.; KANDING, B.; SELVAN, R. Carbontracker: Tracking and predicting the carbon footprint of training deep learning models. **arXiv preprint arXiv:2007.03051**, 2020.

BARHMI, K. et al. A review of solar forecasting techniques and the role of artificial intelligence. In: MDPI. **Solar**. [S.l.], 2024. v. 4, n. 1, p. 99–135.

BELTOZAR-CLEMENTE, S. et al. Predicting customer abandonment in recurrent neural networks using short-term memory. **Journal of Open Innovation: Technology, Market, and Complexity**, Elsevier, v. 10, n. 1, p. 100237, 2024.

BENGIO, Y.; SIMARD, P.; FRASCONI, P. Learning long-term dependencies with gradient descent is difficult. **IEEE transactions on neural networks**, IEEE, v. 5, n. 2, p. 157–166, 1994.

BOMMASANI, R. et al. On the opportunities and risks of foundation models. **arXiv preprint arXiv:2108.07258**, 2021.

CAI, H. et al. Once-for-all: Train one network and specialize it for efficient deployment. **arXiv preprint arXiv:1908.09791**, 2019.

CARUANA, R. et al. Ensemble selection from libraries of models. In: **Proceedings of the twenty-first international conference on Machine learning**. [S.l.: s.n.], 2004. p. 18.

CHO, K. et al. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In: **Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)**. [S.l.: s.n.], 2014. p. 1724–1734.

CHU, Y. et al. A review of distributed solar forecasting with remote sensing and deep learning. **Renewable and Sustainable Energy Reviews**, Elsevier, v. 198, p. 114391, 2024.

COURBARIAUX, M. et al. Binarized neural networks: Training deep neural networks with weights and activations constrained to+ 1 or-1. **arXiv preprint arXiv:1602.02830**, 2016.

DIPIETRO, R.; HAGER, G. D. Chapter 21 - deep learning: Rnns and lstm. In: ZHOU, S. K.; RUECKERT, D.; FICHTINGER, G. (Ed.). **Handbook of Medical Image Computing and Computer Assisted Intervention**. Academic Press, 2020, (The Elsevier and MICCAI Society Book Series). p. 503–519. ISBN 978-0-12-816176-0. Disponível em: https://www.sciencedirect.com/science/article/pii/B9780128161760000260.

DUTTA, A.; KUMAR, S.; BASU, M. A gated recurrent unit approach to bitcoin price prediction. **Journal of risk and financial management**, MDPI, v. 13, n. 2, p. 23, 2020.

ELMAN, J. L. Finding structure in time. **Cognitive science**, Wiley Online Library, v. 14, n. 2, p. 179–211, 1990.

FOLEY, A. M. et al. Current methods and advances in forecasting of wind power generation. **Renewable energy**, Elsevier, v. 37, n. 1, p. 1–8, 2012.

FOROOTAN, M. M. et al. Machine learning and deep learning in energy systems: A review. **Sustainability**, MDPI, v. 14, n. 8, p. 4832, 2022.

HAN, S. et al. Eie: Efficient inference engine on compressed deep neural network. **ACM SIGARCH Computer Architecture News**, ACM New York, NY, USA, v. 44, n. 3, p. 243–254, 2016.

HE, K. et al. Deep residual learning for image recognition. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2016. p. 770–778.

HENDERSON, P. et al. Towards the systematic reporting of the energy and carbon footprints of machine learning. **Journal of machine learning research**, v. 21, n. 248, p. 1–43, 2020.

HINTON, G.; VINYALS, O.; DEAN, J. Distilling the knowledge in a neural network. **arXiv preprint arXiv:1503.02531**, 2015.

HOCHREITER, S.; SCHMIDHUBER, J. Lstm can solve hard long time lag problems. **Advances in neural information processing systems**, v. 9, 1996.

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

JACOB, B. et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In: **Proceedings of the IEEE conference on computer vision and pattern recognition**. [S.l.: s.n.], 2018. p. 2704–2713.

JADERBERG, M.; VEDALDI, A.; ZISSERMAN, A. Speeding up convolutional neural networks with low rank expansions. **arXiv preprint arXiv:1405.3866**, 2014.

KRIKUNOV, A. V.; KOVALCHUK, S. V. Dynamic selection of ensemble members in multi-model hydrometeorological ensemble forecasting. **Procedia Computer Science**, Elsevier, v. 66, p. 220–227, 2015.

LECUN, Y.; DENKER, J.; SOLLA, S. Optimal brain damage. **Advances in neural information processing systems**, v. 2, 1989.

LEE, N.; AJANTHAN, T.; TORR, P. H. Snip: Single-shot network pruning based on connection sensitivity. **arXiv preprint arXiv:1810.02340**, 2018.

LI, L.; ZHU, J.; SUN, M.-T. Deep learning based method for pruning deep neural networks. In: IEEE. **2019 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)**. [S.l.], 2019. p. 312–317.

LI, N.; YU, Y.; ZHOU, Z.-H. Diversity regularized ensemble pruning. In: SPRINGER. **Joint European conference on machine learning and knowledge discovery in databases**. [S.l.], 2012. p. 330–345.

LICIOTTI, D. et al. A sequential deep learning application for recognising human activities in smart homes. **Neurocomputing**, Elsevier, v. 396, p. 501–513, 2020.

LIN, S. et al. Accelerating convolutional networks via global & dynamic filter pruning. In: STOCKHOLM. **IJCAI**. [S.l.], 2018. v. 2, n. 7, p. 8.

LUERS, A. et al. Will ai accelerate or delay the race to net-zero emissions? **Nature**, Nature Publishing Group UK London, v. 628, n. 8009, p. 718–720, 2024.

MONTGOMERY, D. C.; RUNGER, G. C. **Applied statistics and probability for engineers**. 6th. ed. Hoboken, NJ: John Wiley & Sons, 2014.

MUNEER, A. et al. Short term residential load forecasting using long short-term memory recurrent neural network. **International Journal of Electrical & Computer Engineering (2088-8708)**, v. 12, n. 5, 2022.

MUSBAH, H.; ALY, H. H.; LITTLE, T. A. Energy management of hybrid energy system sources based on machine learning classification algorithms. **Electric Power Systems Research**, Elsevier, v. 199, p. 107436, 2021.

NVIDIA. **NVIDIA Tesla T4 GPU Architecture: Powered by NVIDIA Turing**. Santa Clara, CA, 2019.

PFEIFFER, J. et al. Adapterhub: A framework for adapting transformers. In: **Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations**. [S.l.: s.n.], 2020. p. 46–54.

REDDI, V. J. et al. Mlperf inference benchmark. In: IEEE. **2020 ACM/IEEE 47th Annual International Symposium on Computer Architecture (ISCA)**. [S.l.], 2020. p. 446–459.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **nature**, Nature Publishing Group UK London, v. 323, n. 6088, p. 533–536, 1986.

SAADALLAH, A.; PRIEBE, F.; MORIK, K. A drift-based dynamic ensemble members selection using clustering for time series forecasting. In: SPRINGER. **Joint European conference on machine learning and knowledge discovery in databases**. [S.l.], 2019. p. 678–694.

SCHUSTER, M.; PALIWAL, K. K. Bidirectional recurrent neural networks. **IEEE transactions on Signal Processing**, Ieee, v. 45, n. 11, p. 2673–2681, 1997.

SCHWARTZ, R. et al. Green ai. **Communications of the ACM**, ACM New York, NY, USA, v. 63, n. 12, p. 54–63, 2020.

SEN, J.; MEHTAB, S. Long-and-short-term memory (lstm) networksarchitectures and applications in stock price prediction. **Emerging computing paradigms: Principles, advances and applications**, Wiley Online Library, p. 143–160, 2022.

SHARADGA, H.; HAJIMIRZA, S.; BALOG, R. S. Time series forecasting of solar power generation for large-scale photovoltaic plants. **Renewable Energy**, Elsevier, v. 150, p. 797–807, 2020.

SHEHABI, A. et al. 2024 united states data center energy usage report. 2024.

SHIRI, F. et al. Detection of student engagement in e-learning environments using efficientnetv2-l together with rnn-based models. **Journal of Artificial Intelligence**, Tech Science Press, v. 6, p. 85, 2024.

SHIRI, F. M. et al. A comprehensive overview and comparative analysis on deep learning models: Cnn, rnn, lstm, gru. **arXiv preprint arXiv:2305.17473**, 2023.

SHIVAM, K.; TZOU, J.-C.; WU, S.-C. A multi-objective predictive energy management strategy for residential grid-connected pv-battery hybrid systems based on machine learning technique. **Energy Conversion and Management**, Elsevier, v. 237, p. 114103, 2021.

SOMU, N.; MR, G. R.; RAMAMRITHAM, K. A deep learning framework for building energy consumption forecast. **Renewable and Sustainable Energy Reviews**, Elsevier, v. 137, p. 110591, 2021.

TSOUMAKAS, G.; PARTALAS, I.; VLAHAVAS, I. An ensemble pruning primer. In: **Applications of supervised and unsupervised ensemble methods**. [S.l.]: Springer, 2009. p. 1–13.

TURETSKYY, A. et al. Battery production design using multi-output machine learning models. **Energy Storage Materials**, Elsevier, v. 38, p. 93–112, 2021.

VADERA, S.; AMEEN, S. Methods for pruning deep neural networks. **Ieee Access**, IEEE, v. 10, p. 63280–63300, 2022.

VOYANT, C. et al. Machine learning methods for solar radiation forecasting: A review. **Renewable energy**, Elsevier, v. 105, p. 569–582, 2017.

WAQAS, M.; HUMPHRIES, U. W. A critical review of rnn and lstm variants in hydrological time series predictions. **MethodsX**, Elsevier, v. 13, p. 102946, 2024.

WEI, X. et al. Machine learning for pore-water pressure time-series prediction: Application of recurrent neural networks. **Geoscience Frontiers**, Elsevier, v. 12, n. 1, p. 453–467, 2021.

WEIGEND, A. S.; RUMELHART, D. E.; HUBERMAN, B. A. Generalization by weight-elimination applied to currency exchange rate prediction. In: IEEE. **[Proceedings] 1991 IEEE International Joint Conference on Neural Networks**. [S.l.], 1991. p. 2374–2379.

WERBOS, P. J. Generalization of backpropagation with application to a recurrent gas market model. **Neural networks**, Elsevier, v. 1, n. 4, p. 339–356, 1988.

YU, Y. et al. A review of recurrent neural networks: Lstm cells and network architectures. **Neural computation**, MIT Press One Rogers Street, Cambridge, MA 02142-1209, USA journals-info . . . , v. 31, n. 7, p. 1235–1270, 2019.

ZHAO, E. et al. Time-aware maddpg with lstm for multi-agent obstacle avoidance: A comparative study. **Complex & Intelligent Systems**, Springer, v. 10, n. 3, p. 4141–4155, 2024.

ZHOU, Z. et al. Online filter weakening and pruning for efficient convnets. In: IEEE. **2018 IEEE International Conference on Multimedia and Expo (ICME)**. [S.l.], 2018. p. 1–6.

ZHOU, Z. et al. Online filter clustering and pruning for efficient convnets. In: IEEE. **2018 25th IEEE International Conference on Image Processing (ICIP)**. [S.l.], 2018. p. 11–15.