Research, Society and Development, v. 14, n. 6, e3914649003, 2025
(CC BY 4.0) | ISSN 2525-3409 | DOI: http://dx.doi.org/10.33448/rsd-v14i6.49003

Análise preditiva de afastamentos por saúde entre servidores do Tribunal Regional Eleitoral do Rio Grande do Norte utilizando redes neurais LSTM

Predictive analysis of health-related absences among employees of the Regional Electoral Court of Rio Grande do Norte using LSTM neural networks

Análisis predictivo de ausencias por motivos de salud entre funcionarios del Tribunal Regional Electoral de Rio Grande do Norte utilizando redes neuronales LSTM

Recebido: 02/06/2025 | Revisado: 09/06/2025 | Aceitado: 09/06/2025 | Publicado: 13/06/2025

Flávio Roberto Guerra Seabra
ORCID: https://orcid.org/0009-0000-4891-0327
Universidade Federal do Rio Grande do Norte, Brasil
E-mail: flavio.seabra@tre-rn.jus.com
João Carlos Xavier Júnior
ORCID: https://orcid.org/0000-0003-1517-2211
Universidade Federal do Rio Grande do Norte, Brasil
E-mail: jcxavier@imd.ufrn.br

Resumo
A Justiça Eleitoral apresenta como característica ter picos de demanda em anos eleitorais e demanda reduzida em anos não eleitorais, exigindo planejamento estratégico de recursos humanos com movimentações internas de pessoal nos períodos críticos. Dentro desse contexto, este trabalho teve por objetivo predizer afastamentos por motivos de saúde no Tribunal Regional Eleitoral do Rio Grande do Norte (TRE-RN), buscando padrões que auxiliem decisões sobre reforço de pessoal em períodos críticos, através da análise de dados históricos de 15 anos de afastamentos médicos (entre 2010 e 2024). A metodologia incluiu análise de dados de afastamentos com uso de uma rede neural Long Short-Term Memory (LSTM) utilizando a biblioteca PyTorch do Python. Os resultados mostraram a eficácia da LSTM na previsão, apresentando MAE (Erro Absoluto Médio) de 3,14 e RMSE (Raiz do Erro Quadrático Médio) de 4,05, superando o modelo SARIMAX tradicional. A pesquisa contribui para a gestão de recursos humanos no TRE-RN, auxiliando decisões sobre alocação de pessoal em períodos eleitorais, além de fornecer mais subsídios para monitoramento das condições gerais de saúde ocupacional por parte do setor responsável pelos cuidados de saúde dos servidores.
Palavras-chave: Redes Neurais (Computação); Gestão de recursos humanos; Saúde ocupacional; Séries temporais.

Abstract
The Electoral Justice system is characterized by peaks in demand during election years and reduced demand in non-election years, requiring strategic human resource planning with internal staff reassignments during critical periods. In this context, this study aimed to predict health-related absences at the Regional Electoral Court of Rio Grande do Norte (TRE-RN), identifying patterns to support decisions on staff reinforcement during critical periods, based on the analysis of 15 years of medical leave data (from 2010 to 2024). The methodology involved analyzing absence data using a Long Short-Term Memory (LSTM) neural network implemented with the PyTorch library in Python. The results demonstrated the effectiveness of the LSTM model in forecasting, achieving a Mean Absolute Error (MAE) of 3.14 and a Root Mean Square Error (RMSE) of 4.05, outperforming the traditional SARIMAX model. This research contributes to human resource management at TRE-RN by supporting decisions on staff allocation during election periods and providing additional insights for monitoring occupational health conditions by the sector responsible for employee healthcare.
Keywords: Neural Networks (Computer); Human resource management; Occupational health; Time series.

Resumen
La Justicia Electoral se caracteriza por picos de demanda en años electorales y una demanda reducida en años no electorales, lo que exige una planificación estratégica de recursos humanos con reubicaciones internas de personal durante períodos críticos. En este contexto, este trabajo tuvo como objetivo predecir ausencias por motivos de salud en el Tribunal Regional Electoral de Rio Grande do Norte (TRE-RN), buscando patrones que apoyen decisiones sobre el refuerzo de personal en períodos críticos, a través del análisis de datos históricos de 15 años de ausencias médicas (entre 2010 y 2024). La metodología incluyó el análisis de datos de ausencias utilizando una red neuronal Long Short-Term Memory (LSTM) implementada con la biblioteca PyTorch de Python. Los resultados demostraron la eficacia del

modelo LSTM en la predicción, con un Error Absoluto Medio (MAE) de 3,14 y una Raíz del Error Cuadrático Medio (RMSE) de 4,05, superando al modelo SARIMAX tradicional. La investigación contribuye a la gestión de recursos humanos en el TRE-RN, apoyando decisiones sobre la asignación de personal en períodos electorales y proporcionando más información para el monitoreo de las condiciones generales de salud ocupacional por parte del sector responsable del cuidado de la salud de los servidores.
Palabras clave: Redes Neurales (Computación); Gestión de recursos humanos; Salud ocupacional; Series temporales.

1. Introdução

A gestão eficiente de recursos humanos, apesar de constantemente desafiadora, é fundamental para o funcionamento adequado das instituições governamentais, que prestam serviços diretos à população que as mantém por meio dos impostos arrecadados. Faz-se necessário garantir sempre um número adequado de funcionários para atender os usuários de forma eficaz sem, no entanto, que isso resulte em desperdício de recursos públicos. Para tanto, o monitoramento sistemático e contínuo da força de trabalho disponível constitui estratégia fundamental para o alcance desse equilíbrio (Esculápio, 2013).

Uma ferramenta que forneça a previsão da quantidade de funcionários em afastamento médico em períodos específicos além de permitir ao setor responsável pela saúde dos servidores monitorar eventos importantes de saúde pela detecção de aumentos inesperados na quantidade de servidores afastados, pode também auxiliar no planejamento e alocação estratégica de recursos humanos em períodos críticos, de forma a garantir que a instituição tenha sempre pessoal suficiente para manter a qualidade do atendimento ao público, mantendo a eficiência do serviço e garantindo a continuidade dos processos.

O objetivo do presete estudo é predizer afastamentos por motivos de saúde no Tribunal Regional Eleitoral do Rio Grande do Norte (TRE-RN), buscando padrões que auxiliem decisões sobre reforço de pessoal em períodos críticos, através da análise de dados históricos de 15 anos de afastamentos médicos (entre 2010 e 2024). Com esta finalidade, o presente trabalho busca pesquisar se a análise de uma série temporal de afastamentos por motivos de saúde utilizando uma rede neural do tipo Long Short-Term Memory (LSTM) é capaz de prever a quantidade de servidores da Justiça Eleitoral em licença médica em dias específicos.

O artigo está estruturado com as seguintes seções adicionais. A seção 2 apresenta a Metodologia com referencial teórico de apoio à metodologia, incluindo método de obtenção dos dados e pré processamento, conceitos fundamentais de séries temporais, redes neurais e redes LSTM, métricas de avaliação e apresentação e explicação das variáveis exógenas. A seção 3 apresenta resultados e discussão, com análise estatística descritiva, e resultados das predições da rede LSTM com comparação com análise SARIMAX. Por fim, a seção 4 apresenta as considerações finais do estudo, destacando as principais contribuições, limitações encontradas e possíveis direcionamentos para trabalhos futuros.

2. Metodologia

Realizou-se uma pesquisa laboratorial, quantitativa com emprego de séries temporais e aprendizado de máquina (Nascimento et al., 2015; Pereira et al., 2018). A seguir, detalha-se cada etapa, intercalando os procedimentos realizados com o suporte teórico que fundamenta as escolhas metodológicas.

Séries temporais são dados coletados ao longo do tempo em intervalos regulares, com uma ou mais variáveis. Se há apenas uma variável medida em relação ao tempo, a série é univariada, sendo multivariada se há mais de uma variável medida (Brockwell & Davis, 2016). A análise de séries temporais frequentemente visa fazer predições de valores futuros com base em dados históricos da série, sem necessariamente buscar relações de causa e efeito (Brockwell & Davis, 2016; De Paula, Xavier Júnior, & Miranda, 2020).

As séries temporais podem ser estacionárias, quando se desenvolvem ao longo do tempo ao redor de uma média

constante, ou podem apresentar alguma forma de não estacionariedade, com propriedades em constante mudança. Ambas podem ser usadas em modelos de previsão, embora as séries não estacionárias precisem ser convertidas em estacionárias antes de serem modeladas (Auffarth, 2021; Albeladi, Zafar, & Mueen, 2023).

Diversos métodos para verificar a estacionariedade de séries temporais podem ser utilizados como a aplicação de testes de hipóteses com testes t-Student ou de Wilcoxon para comparar as médias de dois períodos diferentes da série ou testes mais específicos como o ADF (Augmented Dickey-Fuller), usado no presente estudo (Auffarth, 2021; Cheung & Lai, 1998).

Tradicionalmente as análises e predições feitas a partir de séries temporais são feitas utilizando-se modelos baseados em modelos Aautoregressivos de Médias Móveis, ou ARMA. Os modelos ARMA são baseados nos modelos AR (Autoregressivos), que utilizam valores passados da série para prever o valor atual, e nos modelos MA (Médias Móveis), que usam erros passados provocados por eventos imprevistos que afetam a série (conhecidos como ruído branco) para ajustar o modelo. O modelo precisa receber 2 parâmetros (p, q), o parâmetro p especifica quantos valores passados da própria série temporal são usados para prever o valor atual e o parâmetro q especifica quantos termos de ruído branco passados são usados para modelar o valor atual. No entanto, como o modelo ARMA(p,q) é adequado para dados não estacionários foi desenvolvido o modelo ARIMA (Autoregressive Integrated Moving Average) com a capacidade de realizar a diferenciação para transformar séries não estacionárias em estacionárias.

No ARIMA, foi inserido o parâmetro d resultando na fórmula ARIMA(p, d, q), que representa a ordem de diferenciação necessária para tornar a série estacionária, onde d = n significa que a série precisou ser diferenciada n vezes para se tornar estacionária, logo, se d = 0 significa que a série já era estacionária. Mas o ARIMA ainda não aborda adequadamente o fator sazonalidade, que são padrões repetitivos em intervalos regulares presentes em algumas séries. Foi então desenvolvido o modelo SARIMA (Seasonal ARIMA) incorporando a diferenciação sazonal. A diferenciação sazonal calcula as diferenças entre observações separadas por um período sazonal (por exemplo, diferenças entre meses de janeiro de anos consecutivos para sazonalidade anual) e a fórmula evoluiu para SARIMA(p, d, q)(P, D, Q)[s], onde (P, Q e D) são semelhantes a (p, q e d), mas com inclusão de um componente sazonal. O parâmetro [s] corresponde à sazonalidade (por exemplo, s = 12 para dados mensais com sazonalidade anual).

Há ainda situações em que o comportamento de uma série temporal pode ser influenciado por variáveis exógenas. Por esse motivo, surgiu o modelo SARIMAX (SARIMA com Regressores Exógenos), onde o modelo SARIMA é calculado levando-se em conta as variáveis exógenas (Brockwell & Davis, 2016).

No presente estudo, por razões que serão detalhadas mais adiante, foi necessário utilizar o modelo SARIMAX uma vez que os dados se apresentaram com padrão se sazonalidade anual e com aparente interferência de fatores exógenos. Para a execução do SARIMAX, o modelo teve o parâmetro [s] determinado em (s=12), devido ao padrão sazonal anual dos dados da série, e os parâmetros (p,d,q) e (P, D, Q) foram escolhidos com a utilização do algoritmo AutoARIMA (Python Software Foundation, 2023), que selecionou automaticamente os valores (3, 0, 3)(2, 0, 0).

Para avaliar e comparar a acurácia das previsões geradas por diferentes métodos ou modelos de análise de séries temporais é importante se utilizar métricas quantitativas. As principais métricas são MAE (Erro Médio Absoluto), MSE (Erro Quadrático Médio), RMSE (Raiz do Erro Quadrático Médio) e MAPE (Erro Percentual Absoluto Médio). O MAE representa a média da magnitude dos erros de previsão. Como é considerado o módulo da subtração entre o valor predito e o valor real, para o índice não importa se o valor predito foi maior ou menor que o valor real. É de fácil interpretação, pois a unidade de medida é a mesma unidade dos dados originais da série. Já o MSE, em vez de fazer o módulo da subtração, eleva esse valor ao quadrado. Com isso fica mais sensível a outliers, porém perde-se em facilidade de interpretação, pois a unidade de medida também é elevada ao quadrado. O RMSE é a aplicação da raiz quadrada no valor da MSE, retomando a normalidade da unidade de medida. O MAPE expressa a acurácia da previsão como uma porcentagem do valor real. Mas como no cálculo o

valor da subtração do valor predito pelo valor real precisa ser dividido pelo valor real, o MAPE não é indicado se existe na série valores reais iguais ou próximos a zero (Brockwell & Davis, 2016). Os cálculos das métricas estão demonstradas no Quadro 1.

Quadro 1 - Métricas de Avaliação Utilizadas

(p_i é o valor predito pela análise da série e a_i é o valor real)

Nome da Métrica	***Definição***
Erro absoluto médio	$MAE = \frac{1}{n}\sum_{i=1}^{n}\lvert p_i - a_i\rvert$
Erro Quadrático Médio	$MSE = \frac{1}{n}\sum_{i=1}^{n}(p_i - a_i)^2$
Raiz do Erro Quadrático Médio	$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(p_i - a_i)^2}$
Erro Percentual Absoluto Médio	$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left\lvert\frac{p_i - a_i}{a_i}\right\rvert$

Fonte: Flávio Seabra (2025).

No presente estudo, considerando a impossibilidade de se usar a métrica MAPE, uma vez que na série temporal existe valores zero e optando-se por melhor interpretabilidade dos resultados, foram utilizadas as métricas MAE e RMSE.

Redes Neurais (RN) surgiram da busca por sistemas computacionais com processamento semelhante ao cérebro biológico. Propostas na década de 1940 com o modelo de neurônio artificial de McCulloch e Pitts e o Perceptron de Rosenblatt, ganharam força nos anos 1980 com modelos matemáticos mais sólidos possibilitados por hardwares mais robustos, demonstrando seu potencial para resolver problemas complexos e impulsionando desenvolvimentos futuros (Fleck et al., 2016; Faceli et al., 2023).

Nas redes neurais artificiais, os neurônios são unidades de processamento interconectadas em camadas de entrada (dados), saída (resultados) e, opcionalmente, camadas intermediárias (também chamadas de camadas ocultas). Redes com poucas camadas são classificadas como rasas, enquanto redes com muitas camadas são chamadas de redes profundas. Podem ser totalmente conectadas, onde cada neurônio está ligado a todos da camada seguinte, ou parcialmente conectadas, com conexões específicas entre neurônios (Faceli et al., 2023).

No processo chamado *feedforward*, os neurônios da camada de entrada recebem dados brutos e os repassam, multiplicados por um peso sináptico *w* aleatório. Nas camadas seguintes, cada neurônio recebe os valores da camada anterior, multiplica-os por seus respectivos pesos sinápticos, soma um valor de Bias *b*, aplica uma função de ativação *f* e repassa o resultado com um novo peso sináptico para a próxima camada, repetindo o processo, até a saída da rede (Faceli et al., 2023; Mesquita, 2024). No treinamento da rede neural, os valores de saída são então comparados ao valor real e a diferença entre eles é propagada de volta pela rede em um processo chamado *backpropagation*. Neste processo, utiliza-se um algoritmo de otimização chamado gradiente descendente, que ajusta iterativamente os pesos para minimizar o erro, calculando derivadas parciais da função de erro em relação a cada peso e atualizando-os proporcionalmente à sua contribuição para o erro total, permitindo que a rede gradualmente "aprenda" o padrão dos dados.

As Redes Neurais Recorrentes (RNNs, do inglês Recurrent Neural Networks) foram criadas para lidar com problemas onde é importante lembrar de dados anteriores, pois as redes neurais até então não guardavam estes dados. Ao contrário das redes tradicionais, as RNNs têm ciclos entre suas unidades, permitindo que a informação flua em múltiplos sentidos gerando uma memória de curto prazo. Isso torna as RNNs adequadas para problemas com dados sequenciais, como traduções,

processamento de linguagem natural e reconhecimentos de fala, situações em que a saída depende tanto da entrada corrente quanto de entradas anteriores (Hochreiter & Schmidhuber, 1997; Mesquita, 2024). Nessas redes, além das unidades de entrada, intermediária e de saída, há também unidades de contexto, que não interagem com o ambiente externo, e são usadas apenas para memorizar as ativações anteriores das unidades intermediárias. Durante o treinamento da rede com o *backpropagation*, as ativações das unidades intermediárias recebem, além dos valores do neurônio da camada anterior, os valores das unidades de contexto, de forma recorrente (Härter, 2007).

No entanto, problemas complexos, como séries temporais, exigem muitas camadas ocultas (redes profundas) e a capacidade de reter informações sequenciais das primeiras RNNs é limitada a alguns poucos passos anteriores porque nesses casos, o *backpropagation*, pode sofrer com o problema do *vanishing gradient*, pois o gradiente descendente faz os valores dos pesos serem reduzidos exponencialmente ao retroceder pelas camadas chegando muito próximo de 0, impedindo o aprendizado da rede. Em RNNs tradicionais, mesmo com o algoritmo *backpropagation through time* (BPTT), uma evolução do *backpropagation* para permitir a adição de mais camadas na rede sem que o problema do *vanishing gradient* aconteça, observou-se que o problema persiste para redes com mais de 10 camadas (Rostamian & O'Hara, 2022).

Para resolver o problema do *vanishing gradient* das RNNs, Hochreiter e Schmidhuber (1997) propuseram o modelo Long Short-Term Memory (LSTM). As LSTMs possuem células de memória com "portões" com a capacidade de adicionar, descartar e atualizar as informações no tempo, controlando assim o fluxo de informações. Essa arquitetura permite que o sinal de erro se propague no tempo sem decaimento, resolvendo o *vanishing gradient* e permitindo que a rede aprenda e memorize informações por longos períodos, mesmo em sequências com mais de 1000 passos (Spancerski & Santos, 2021). O conceito de Constant Error Carousel (CEC), que hoje é chamado de estado da célula (C_t), foi descrito como um dos mecanismos-chave que ajudam a resolver o problema do vanishing gradient porque permite que os gradientes sejam mantidos constantes ao longo do tempo durante o processo de *backpropagation* (Hochreiter & Schmidhuber, 1997).

Na arquitetura original proposta por Hochreiter e Schmidhuber (1997), uma unidade multiplicativa chamada portão de entrada foi introduzida para controlar quais novas informações deviam ser adicionadas ao estado da célula e uma unidade multiplicativa chamada portão de saída controlava quanto da informação armazenada seria usada para a próxima saída. Em 2000 foi introduzido o portão de esquecimento, que permitiu que as redes LSTM descartem dados que considerem desnecessários (Gers, Schraudolph & Schmidhuber, 2002).

Explicando mais detalhadamente, como a LSTM é uma rede recorrente, uma única célula LSTM é suficiente para processar um conjunto de dados em sequência, como é característico das séries temporais. O processamento se inicia com três valores entrando na célula LSTM: um valor chamado C_{t-1}, um valor chamado H_{t-1}, ambos provenientes da passagem anterior pela célula LSTM e um valor chamado X_t , ou Entrada. Como o processamento está iniciando, não existe valor proveniente de passagem anterior pela LSTM, por isso os valores iniciais de C_{t-1} e H_{t-1} são pré-estabelecidos como zero e são fornecidos à rede no ato da implementação. Já o primeiro valor X_t é o primeiro valor da série temporal (Gers, Schraudolph & Schmidhuber, 2002; Livieris, Pintelas & Pintelas, 2020; Rostamian & O'Hara 2022).

O processamento dentro da célula LSTM se inicia no portão de esquecimento da seguinte forma: o valor H_{t-1} é multiplicado pelo seu respectivo peso W^h_f, o valor X_t é multiplicado pelo seu respectivo peso W^x_f , depois os resultados destas multiplicações são somados e esse resultado é somado ao valor Bias (b_f). Depois esse resultado passa por uma função de ativação sigmoide, resultando no valor f_t que é chamado valor do Portão de Esquecimento (f_t). Logo, o valor gerado pelo portão de esquecimento (f_t) é definido como $f_t = \sigma\,(W^h_f\, H_{t-1} + W^x_f\, X_t + b_f)$, e será entre 0 e 1. O valor f_t será então multiplicado pelo valor que entrou na célula LSTM como C_{t-1} (estado da célula anterior), na forma ($f_t \odot C_{t-1}$). Isso significa que se o valor f_t for mais próximo de 0, a célula LSTM estará "esquecendo" maior parte do C_{t-1}, se for mais próximo de 1, mais do C_{t-1} será "lembrado", sendo considerado para as etapas seguintes (Gers, Schraudolph & Schmidhuber, 2002; Livieris, Pintelas &

Pintelas, 2020; Rostamian & O'Hara 2022).

Paralelamente, o portão de entrada está fazendo duas operações simultâneas. A primeira é o cálculo do valor chamado Estado Candidato ($\check{C}_t$), que é feito de forma semelhante ao que foi descrito anteriormente no portão de esquecimento, com as ressalvas de que os valores dos pesos (*W*) e Bias (*B*) são outros, específicos para essa operação (W^h_C , W^x_C e b_C), e de que a função de ativação, nesse caso, não é a função sigmoide e sim a função tangente hiperbólica, que gera valores entre -1 e 1. Logo, o cálculo do valor candidato é definido como $\check{C}_t = \tanh (W^h_C H_{t-1} + W^x_C X_t + b_C)$. A segunda operação feita pelo portão de entrada, para gerar o valor i_t, que é o Valor do Portão de Entrada, é bem semelhante à feita no portão de esquecimento, ou seja, $i_t = \sigma (W^h_i H_{t-1} + W^x_i X_t + b_i)$. Depois de gerados os valores $\check{C}_t$ e i_t são multiplicados entre si ($\check{C}_t \odot i_t$) e o valor resultante é somado ao valor proveniente do portão de esquecimento ($f_t \odot C_{t-1}$), resultando no valor C_t (que nas primeiras publicações de LSTM era conhecido como CEC). Logo, C_t é o valor que contém a informação que é o próprio estado atual da célula LSTM e é definido por $C_t = f_t \odot C_{t-1} + \check{C}_t \odot i_t$ (Gers, Schraudolph & Schmidhuber, 2002; Livieris, Pintelas & Pintelas, 2020; Rostamian & O'Hara 2022).

Nesse momento o valor C_t segue dois caminhos distintos. No primeiro, ele vai compor a memória de longo prazo, saindo inalterado da célula LSTM para entrar novamente, de forma recorrente, já como C_{t-1} para o próximo processamento. No segundo caminho, o valor C_t passará pelo portão de saída para gerar o valor H_t. Para isso, o valor C_t passa por uma função de ativação *tanh* para restringir seus valores entre -1 e 1 e o resultado é multiplicado pelo valor o_t , que é o Valor do Portão de Saída, e é formado através de uma função de ativação sigmoide de forma semelhante aos valores f_t e i_t explicados anteriormente. Logo, $o_t = \sigma (W^h_o H_{t-1} + W^x_o X_t + b_i)$ e $H_t = o_t \odot \tanh(C_t)$. Depois que o valor H_t é formado, ele sai da célula LSTM como memória de curto prazo para entrar novamente, de forma recorrente, para o próximo processamento como valor H_{t-1} (Gers, Schraudolph & Schmidhuber, 2002; Livieris, Pintelas & Pintelas, 2020; Rostamian & O'Hara 2022).

Em casos onde a rede LSTM possui mais de uma camada, como é o caso do presente trabalho por motivos que serão explicados mais adiante, o valor H_t, além de entrar como H_{t-1} na mesma camada, entra também como X_t na camada LSTM seguinte.

Estudos recentes demonstram a aplicabilidade e o desempenho superior das redes neurais recorrentes do tipo LSTM na modelagem e previsão de séries temporais em diversos domínios, com potencial para aplicações na área da saúde. Nelson (2017) utilizou dados históricos da BM&FBovespa para prever tendências de preços, demonstrando que o modelo LSTM superou algoritmos tradicionais como MLP e Random Forest em métricas de acurácia, precisão, revocação, medida F1 e indicadores financeiros. Spancerski e Santos (2021) aplicaram LSTM à previsão da produtividade de arroz no Rio Grande do Sul, obtendo erro percentual absoluto médio (MAPE) inferior a 1%, indicando excelente desempenho. Dubey et al. (2021) compararam LSTM com ARIMA e SARIMA na previsão do consumo diário de energia, obtendo melhor resultado com o LSTM (RMSE = 0,23). De forma semelhante, Sirisha, Belavagi e Attigeri (2022) observaram desempenho superior do LSTM na previsão de lucros em vendas (MAE = 3,26; RMSE = 3,92), em comparação aos modelos ARIMA e SARIMA. Zha et al. (2022) utilizaram um modelo híbrido CNN-LSTM para estimar a produção de gás e água em reservatórios, destacando sua superioridade nas fases estável e de declínio. Esses resultados sugerem que modelos LSTM podem ser adaptados com êxito a cenários da saúde pública e ocupacional, como a previsão de afastamentos, hospitalizações ou demandas por serviços, contribuindo para o planejamento e a gestão eficientes de recursos.

Foi obtido a partir do módulo de afastamentos médicos do sistema SGRH do TRE-RN um relatório em formato CSV de todos os afastamentos do tipo "LICENÇA PARA TRATAMENTO DA PRÓPRIA SAÚDE" dos servidores do Tribunal entre Janeiro de 2010 e Dezembro de 2024. Em seguida os dados foram pré processados utilizando a biblioteca Python Pandas (The Pandas Development Team, 2020) até a obtenção de um DataFrame onde cada linha (instância) corresponde a uma data

entre 01/01/2010 e 31/12/2024 (com um total de 5480 linhas), e uma coluna contendo a quantidade de funcionários do TRE-RN que estavam de licença médica naquela data específica, caracterizando uma série temporal univariada.

A análise LSTM foi executada em um notebook em ambiente Google Colab configurado com GPU A100, utilizando a biblioteca para aprendizado profundo PyTorch (Meta AI, 2023), e o modelo foi configurado para trabalhar com 80% de dados de treinamento e 20% de dados de teste. Os parâmetros *seq_length, num_epochs, num_layers, hidden_size* e *batch_size* da LSTM foram incluídos em um *grid search* para serem testados com diferentes valores (Quadro 2).

Quadro 2 - Parâmetros da LSTM que foram testados em *grid Search.*

Parâmetro	Valores	Descrição
seq_length	100 e 1000	Define o tamanho da janela de dados sequenciais que a célula LSTM processa simultaneamente, recebendo seq_length valores de cada vez em janelas deslizantes de passo 1, como vetor X
batch_size	128 e 256	Determina quantas janelas deslizantes de tamanho seq_length serão processadas simultaneamente em paralelo nas camadas LSTM.
hidden_size	128 e 256	Tamanho do vetor H_t , que corresponde à saída da primeira camada LSTM e entrada das camadas seguintes, se houver.
num_layers	1, 2 e 3	Número de camadas LSTM
num_epochs	100 e 200	Número de vezes que o modelo LSTM percorrerá todo o conjunto de dados de treinamento.
Remoção de outliers	Ativada ou Desativada	Função que remove os dados que estejam fora do intervalo interquartil antes dos dados entrarem na LSTM
Remoção de variáveis exógenas	Ativada ou Desativada	Função que remove as variáveis exógenas antes dos dados entrarem na LSTM

Fonte: Flávio Seabra (2025).

Foram incluídas na análise três variáveis exógenas. A primeira variável exógena marcou todos os dias entre 01/03/2020 e 31/12/2021, por considerar razoável que pode ter havido alguma influência da pandemia de COVID-19 no padrão de afastamentos por motivos de saúde nesse período. As outras duas variáveis exógenas, uma para diferenciar os meses de recesso judiciário dos demais meses do ano e outra para diferenciar anos eleitorais de anos não eleitorais serão explicadas no próximo capítulo.

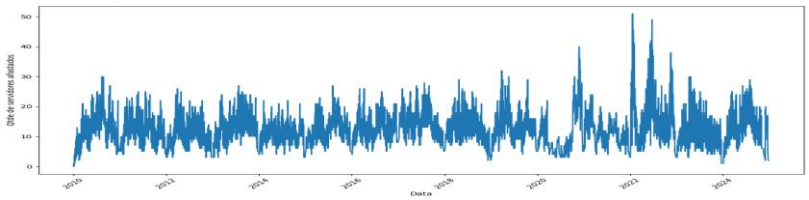
Uma função para remover *outliers* e outra para remover as variáveis exógenas antes dos dados entrarem na LSTM foram construídas e incluídas no *grid search* para que estas duas condições fossem testadas.

Considerando todas as variáveis testadas em grid search, a combinação de parâmetros resultou em um total de 192 execuções. Em todas elas os parâmetros MAE e RMSE foram obtidos para identificar qual a melhor combinação de parâmetros.

3. Resultados e Discussão

A Figura 1 mostra graficamente a série temporal dos afastamentos por dia ao longo do período analisado.

Figura 1 - Gráfico de série temporal das quantidades de servidores afastados por dia.



Fonte: Flávio Seabra (2025).

É possível visualizar grande variabilidade, com picos e vales que indicam momentos de maior e menor quantidade de servidores afastados, que parece flutuar em torno de um valor médio, sem tendência clara de crescimento ou decrescimento ao longo do período.

Para identificar se a série testada apresentava estacionariedade, foi executado o teste ADF, cujos resultados encontram-se na Tabela 1.

Tabela 1 - Resultados do Augmented Teste de Dickey-Fuller (ADF).

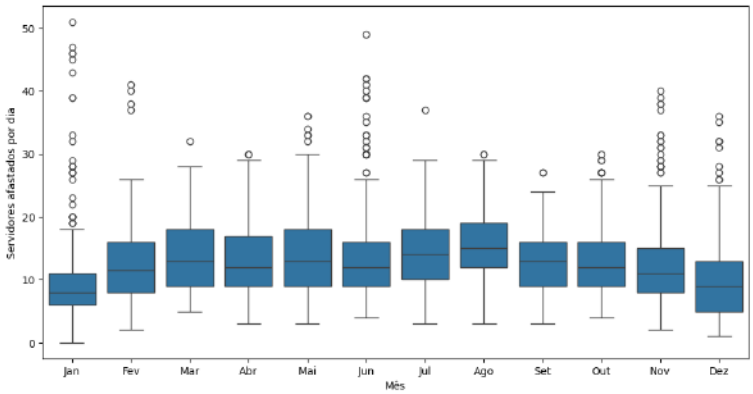
Métrica	**Valor**
ADF Statistic:	−8.41075
p-value:	2.10709e−13

Fonte: Flávio Seabra (2025).

O teste ADF parte da hipótese nula H0 de que a série temporal possui uma raiz unitária e não é estacionária e então calcula um p-value. Se p > 0, 05, a hipótese H0 não pode ser rejeitada, o que significa que a série é do tipo não estacionária. Se p < 0, 05, a hipótese pode ser rejeitada, o que indica estacionariedade da série, o que significa que a série testada no presente estudo é uma série temporal estacionária.

É possível observar já na análise visual da Figura 1 que os valores são mais baixos durante os meses iniciais e finais que nos outros meses de cada ano. As Figuras 2 e 3 mostram esse fenômeno de forma mais específica.

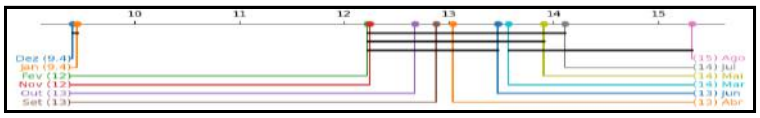
Figura 2 - Boxplot dos afastamentos de acordo com os meses do ano.



Fonte: Flávio Seabra (2025).

Figura 3 - Diagrama de diferenças críticas entre os meses, evidenciando diferença estatisticamente significante entre

os meses de Janeiro e Dezembro em relação aos demais meses.



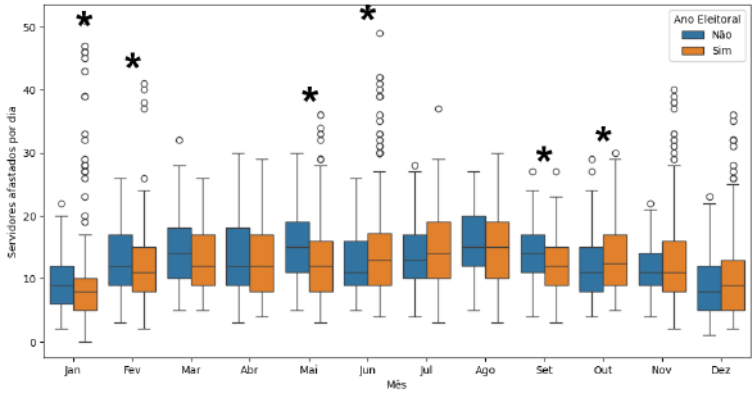
Fonte: Flávio Seabra (2025).

As Figuras 2 e 3 mostram que, de acordo com o teste estatístico de Friedman, os meses de Janeiro e Dezembro apresentam valores estatisticamente semelhantes de mediana de servidores afastados por dia e significantemente diferentes dos demais meses do ano. Como considerou-se que essa diferença provavelmente se deve ao recesso judiciário, condição em que não há expediente do dia 20 de dezembro de um ano ao dia 06 de janeiro do ano seguinte, entendeu-se que o recesso judiciário deve ser levado em conta como uma variável exógena.

Como na justiça eleitoral os anos pares são os anos em que tradicionalmente são feitas as eleições, alternando entre eleições municipais e gerais, foi verificado se houve diferença significante no padrão de afastamentos entre anos eleitorais e não eleitorais para determinar se o fator ano eleitoral ou não eleitoral deveria ser considerado como uma variável exógena (Figura 4).

Figura 4 - Boxplots da quantidade de servidores afastados separados por mês em todo o período da série temporal, comparando-se anos eleitorais e não eleitorais.

* Diferença estatisticamente significante (Teste Mann-Whitney p< 0,05).

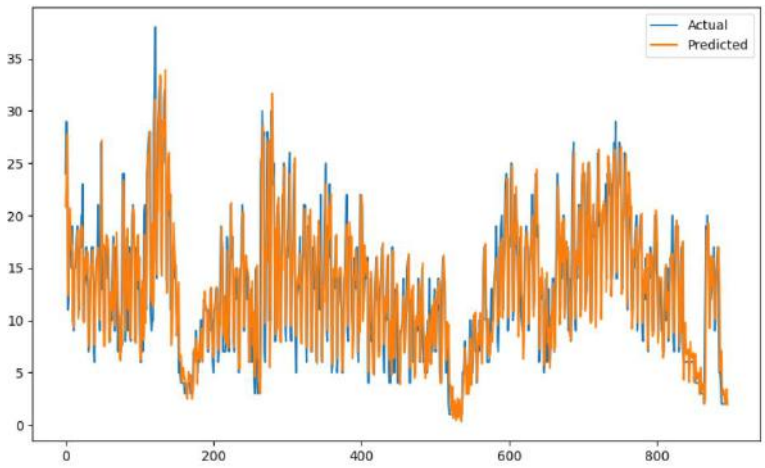


Fonte: Flávio Seabra (2025).

Observou-se diferenças estatisticamente significantes entre anos eleitorais e anos não eleitorais de forma marcante em meses coincidentes ou próximos de eventos importantes nos anos eleitorais, como o mês de Maio, no qual se encontra o final do prazo de alistamento e transferências de domicílio eleitoral, o mês de Setembro, que é o primeiro mês que se encontra inteiramente dentro do período eleitoral (que se inicia em meados do mês de Agosto) e o mês de Outubro, onde tradicionalmente de realiza o primeiro turno das eleições. Justifica-se então considerar ano eleitoral e não eleitoral como uma variável exógena na análise.

Das 192 execuções da rede LSTM, a que apresentou melhores resultados de acordo com as métricas utilizadas no presente estudo apresentou MAE = 2,12 e RMSE = 3,07. A Figura 5 mostra o gráfico comparativo dos valores preditos pela análise LSTM superpostos aos valores reais no conjunto de dados de teste.

Figura 5 - Gráfico com os valores reais e os valores preditos pelo LSTM.



Fonte: Flávio Seabra (2025).

Observa-se que a adesão dos valores preditos (laranja) aos valores reais (azul) no gráfico pode ser considerada boa, indicando que o modelo capturou eficazmente o padrão dos dados, embora tenha apresentado alguma dificuldade em prever valores extremos.

O Quadro 3 mostra os valores dos parâmetros que pela análise no *grid search* foram os que resultaram nos menores valores das métricas MAE e RMSE.

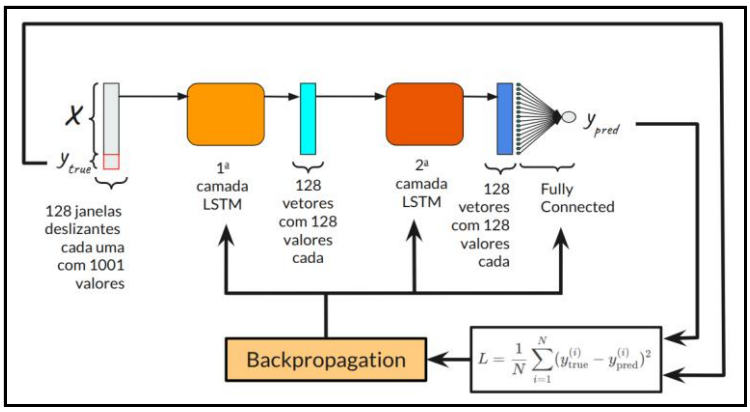
Quadro 3 - Parâmetros da LSTM que resultaram nos menores valores de MAE e RMSE.

Parâmetro	**Valores**
seq_length	1000
batch_size	128
hidden_size	128
num_layers	2
num_epochs	100
Remoção de outliers	Desativada
Remoção de variáveis exógenas	Ativada

Fonte: Flávio Seabra (2025).

Com os parâmetros mostrados no Quadro 3, cada uma das 100 épocas da análise se processou como se segue (Figura 6).

Figura 6 - Descrição do fluxo de cada batch da análise LSTM em cada época.



Fonte: Flávio Seabra, 2025

Como a divisão para a análise foi configurada com 80% para treinamento e 20% para teste e um *DataFrame* de 5480 linhas, o conjunto de treinamento foi formado pelas primeiras 4384 datas e um conjunto de teste com as últimas 1096 datas. Como foi utilizado *seq_length* de 1000, foram criadas 3384 janelas deslizantes com passo 1 para o treinamento (4384 - 1000 = 3384). Cada janela é composta por 1000 valores, formando o conjunto X (os primeiros 1000 valores da janela), com o 1001º valor servindo como o rótulo y_{true}. Dessa forma, a célula LSTM utiliza os 1000 valores de cada janela para predizer o valor imediatamente seguinte.

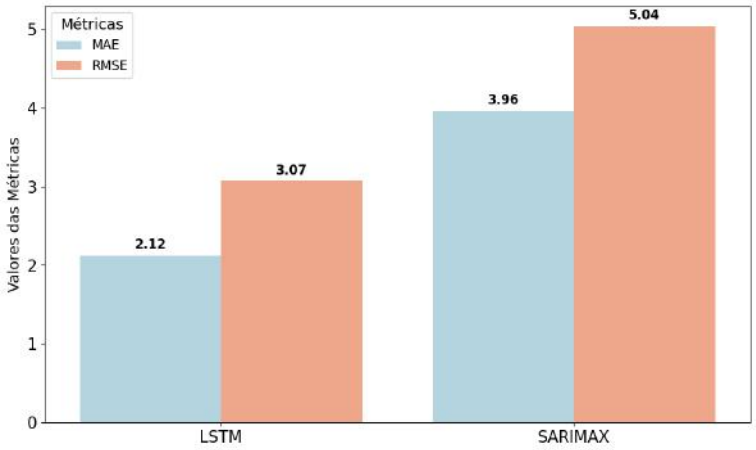
Com um *batch_size* de 128, o treinamento foi dividido em 27 *batches*. Vinte e seis desses *batches* contiveram 128 janelas deslizantes, e o último *batch* ficou com as 96 janelas restantes (totalizando 3384 janelas = 26 · 128 + 96).

Por termos *num_layers* de 2, eram configuradas duas camadas LSTM. Na primeira camada, em cada um dos 26 *batches*, 128 das 3384 janelas deslizantes com 1000 valores eram carregadas em 128 células LSTM. Como resultado do processamento da primeira camada, cada célula LSTM produzia um vetor H_t de 128 valores (porque *hidden_size* = 128). Cada um destes vetores era entregue a uma das 128 células LSTM da segunda camada, que também produzia como resultado do processamento outro vetor H_t com 128 valores. O vetor resultante da saída de cada célula LSTM da segunda camada era então entregue a uma rede neural *fully connected* que, finalmente, gerava o valor predito y_{pred} referente àquela janela de 1000 valores.

O valor y_{pred} era então comparado ao valor y_{true} de sua janela deslizante correspondente, com uma função de perda quantificando o erro e gerando valores para a *backpropagation* ajustar os pesos e *biases* das duas camadas LSTM e da rede *fully connected*. Após os ajustes na *backpropagation* o processamento do próximo *batch* se iniciava com as próximas 128 janelas deslizantes, repetindo-se esse ciclo até que todos os *batches* fossem processados pela rede.

Para avaliar se as predições feitas pela LSTM podem ser consideradas bons indicadores, a análise com o método SARIMAX foi também executada, incorporando o padrão de sazonalidade observado (valores reduzidos em janeiro e dezembro) e as variáveis exógenas identificadas na análise estatística. A análise SARIMAX teve métricas de desempenho de MAE = 3,96 e RMSE = 5,04. A Figura 7 mostra um gráfico comparativo com as métricas MAE e RMSE dos modelos LSTM e do modelo SARIMAX.

Figura 7 - Gráfico com as métricas MAE e RMSE dos modelos LSTM e do modelo SARIMAX.



Fonte: Flávio Seabra (2025).

A comparação dos valores das métricas mostrou superioridade na qualidade preditiva da rede LSTM em relação ao modelo SARIMAX considerando as duas métricas utilizadas.

4. Considerações Finais

O presente trabalho teve como objetivo desenvolver e avaliar a aplicação de uma rede neural Long Short-Term Memory (LSTM) para a previsão de afastamentos por motivos médicos de servidores do Tribunal Regional Eleitoral do Rio Grande do Norte (TRE-RN) utilizando dados de uma série temporal. A meta principal foi buscar uma ferramenta que auxilie na gestão estratégica de recursos humanos, particularmente no planejamento de força de trabalho durante os períodos de maior demanda operacional, notadamente nos anos eleitorais, bem como monitoramento de condições de saúde.

Com base nas métricas de desempenho utilizadas no estudo, a análise da série temporal de afastamentos médicos utilizando rede LSTM demonstrou ser superior em relação ao modelo estatístico SARIMAX, com Erro Absoluto Médio (MAE) de 2,12 e Raiz do Erro Quadrático Médio (RMSE) de 3,07, em comparação com os valores de MAE = 3,96 e RMSE = 5,04 obtidos com o uso do método SARIMAX. No entanto, como esperado em séries temporais de alta variabilidade, o modelo apresentou limitações na previsão precisa de valores extremos (outliers).

A otimização via grid search permitiu identificar os parâmetros da rede LSTM que resultaram em melhor generalização do padrão dos dados históricos, inclusive a ausência do impacto das variáveis exógenas na performance da rede.

A contribuição prática deste trabalho reside no fornecimento de previsões mais acuradas sobre quantidade de servidores em licenças médicas em períodos específicos do ano, informação estratégica para a área de gestão de pessoas do TRE-RN. Assim, as previsões feitas com essa ferramenta poderão fazer parte dos parâmetros analisados pelos gestores de recursos humanos no planejamento e tomada de decisão no dimensionamento da força de trabalho e reforços de pessoal principalmente em períodos críticos, auxiliando na manutenção da eficiência operacional e da qualidade dos serviços prestados pela Justiça Eleitoral.

Importante salientar que o estudo apresentou uma limitação relevante que foi a impossibilidade, devido ao conjunto de dados utilizado, de se fazer predições individualizadas em relação ao local de lotação dos servidores, se na capital ou nas zonas eleitorais do interior do estado.

Nesse sentido, configura-se como linha necessária de investigação futura o desenvolvimento de modelos específicos para cada grupo de servidores, com o objetivo de gerar previsões mais ajustadas à realidade local e fornecer subsídios mais

precisos para a gestão de pessoal nas diferentes unidades da Justiça Eleitoral.

Embora o presente estudo valide o potencial das redes LSTM como ferramenta robusta para previsão em séries temporais no contexto da administração pública, estudos futuros com aplicação de arquiteturas mais avançadas de redes neurais, como modelos baseados em Transformers ou estruturas híbridas do tipo CNN-LSTM devem ser testados como alternativa, tendo em vista a complexidade desse tipo de dado, objetivando a mitigação de erros em valores extremos.

Referências

Albeladi, K., Zafar, B., & Mueen, A. (2023). Time series forecasting using LSTM and ARIMA. *International Journal of Advanced Computer Science and Applications*, *14*(1), 313–314. https://www.ijacsa.thesai.org

Auffarth, B. (2021). Machine learning for time-series with Python. Packt Publishing.

Brockwell, P. J., & Davis, R. A. (2016). *Introduction to time series and forecasting* (3rd ed.). Springer.

Cheung, Y.-W., & Lai, K. S. (1998). Power of the augmented Dickey-Fuller test with information-based lag selection. *Journal of Statistical Computation and Simulation, 60*(1), 57–65.

De Paula, D. M., Xavier Júnior, J. C., & Miranda, K. F. (2020). Aplicação de séries temporais para previsão de despesas de energia elétrica do Tribunal Regional Eleitoral do Rio Grande do Norte. *Brazilian Journal of Development, 6*(11), 87089–87112.

Dubey, A. K., Kumar, A., García-Díaz, V., Sharma, A. K., & Kanhaiya, K. (2021). Study and analysis of SARIMA and LSTM in forecasting time series data. *Sustainable Energy Technologies and Assessments, 47*, 101474.

Esculápio, M. (2013). *A gestão de recursos humanos no serviço público* [Monografia]. Universidade Tecnológica Federal do Paraná.

Faceli, K., Lorena, A. C., Gama, J., Almeida, T. A., & Carvalho, A. C. P. L. F. (2023). *Inteligência artificial: Uma abordagem de aprendizado de máquina* (2a ed.). LTC.

Fleck, L., Tavares, M. H. F., Eyng, E., Helmann, A. C., & Andrade, M. A. (2016). Redes neurais artificiais: Princípios básicos. *Revista Eletrônica Científica Inovação e Tecnologia, 7*(15), 47.

Gers, F. A., Schraudolph, N. N., & Schmidhuber, J. (2002). Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research, 3*, 115-143.

Härter, F. P. (2007). *Redes neurais recorrentes aplicadas à assimilação de dados em dinâmica não-linear* [Tese de doutorado, Instituto Nacional de Pesquisas Espaciais].

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735-1780 .

Livieris, I. E., Pintelas, E., & Pintelas, P. (2020). A CNN–LSTM model for gold price time-series forecasting. *Neural Computing and Applications, 32*(23), 17351–17360.

Mesquita, L. A. F. (2024). *Redes neurais artificiais aplicadas a séries temporais para predição de enchentes* [Master's thesis, Instituto de Ciências Matemáticas e de Computação – ICMC-USP].

Meta AI. (2023). *PyTorch* (Version 2.0.0) [Computer software]. https://pytorch.org

Nascimento, E. G. S. et al. (2015). Um algoritmo baseado em técnicas de agrupamento para detecção de anomalias em séries temporais . In: Shitsuka, R. & Shitsuka, D. M. (2015). *Estudos e Práticas de Aprendizagem de Matemática e Finanças com Apoio de Modelagem*. Editora Ciência Moderna

Nelson, D. M. Q. (2017). *Uso de redes neurais recorrentes para previsão de séries temporais financeiras* [Master's thesis, Universidade Federal de Minas Gerais].

Pereira, A. S., Shitsuka, D. M., Parreira, F. J., & Shitsuka, R. (2018). Metodologia da pesquisa científica [Recurso eletrônico]. Santa Maria, RS: UAB/NTE/UFSM.

Python Software Foundation. (2023). *pmdarima: ARIMA modeling with auto-tuning* (Version 2.0.3) [Python package]. https://pypi.org/project/pmdarima/

Rostamian, A., & O'Hara, J. G. (2022). Event prediction within directional change framework using a CNN-LSTM model. *Neural Computing and Applications, 34*(20), 17193-17205 .

Sirisha, U. M., Belavagi, M. C., & Attigeri, G. (2022). Profit prediction using ARIMA, SARIMA and LSTM models in time series forecasting: A comparison. *IEEE Access, 10*, 124715-124727.

Spancerski, J. S., & Santos, J. A. A. D. (2021). Previsão da produtividade de arroz: Uma aplicação de redes neurais recorrentes LSTM. *Revista Cereus, 13*(2), 45-62.

The Pandas Development Team. (2020). *pandas: Python data analysis library* (Version 1.1.2) [Computer software]

Zha, W., Liu, Y., Wan, Y., Luo, R., Li, D., Yang, S., & Xu, Y. (2022). Forecasting monthly gas field production based on the CNN-LSTM model. *Energy, 260*, 124889.