

# Neural Network Models as Proxy for Reservoir Simulators

Johnattan D. F. Viana  
CIn - Centro de Informática  
UFPE  
Recife, Brasil  
jdfv@cin.ufpe.br

Juan A. R. Tueros  
CIn - Centro de Informática  
UFPE  
Recife, Brasil  
juan.tueros@ufpe.br

Victor B. Silva  
CIn - Centro de Informática  
UFPE  
Recife, Brasil  
vbs3@cin.ufpe.br

**Abstract**—Management oil reservoirs involves conducting numerous simulations to understand the field, which results in high computational costs. These costs have a detrimental effect on activities related to controlling production and injection wells for optimizing production. The aim of this study is to develop a neural network-based proxy model using only production history data to replace the reservoir simulator. The proxy model should accurately predict reservoir responses, such as time series of fluid phase flow rates or bottom hole pressure, based on given controls of injection and production wells. For this purpose, in this paper we apply four neural network strategies to capture reservoir dynamics: Long Short Term Memory (LSTM), LSTM with an attention mechanism (LSTM-A), Gated Recurrent Unit (GRU), and GRU with an attention mechanism (GRU-A). To assess the accuracy and precision of the proposed techniques, a synthetic reservoir model from the literature with 4 production wells and 8 injection wells was used. The results indicate that the GRU-A demonstrated superior accuracy and exhibited lower variability compared to other approaches, achieving improvements of up to 13% in reducing RMSE in the test set. The source code is provided<sup>1</sup>.

**Index Terms**—Proxy, Simulador, Séries Temporais, Redes Neurais, Mecanismo de Atenção.

## I. INTRODUCTION

Um reservatório de petróleo é caracterizado pela interação de propriedades geofísicas, geoquímicas e geológicas. Para entender seu comportamento, utiliza-se a simulação de um modelo computacional simplificado que o represente [1]. A simulação envolve a construção de uma malha de elementos de volumes finitos, incorporando determinadas propriedades petrofísicas.

As entradas na simulação são séries temporais de pressão de fundo de poço (BHP) dos injetores e produtores, resultando em respostas como séries temporais de vazões de água, óleo e gás para cada poço produtor, e séries temporais de vazão de injeção para cada poço injetor [2]. As saídas incluem séries temporais de BHP para cada poço injetor, quando fornecidas as séries temporais das vazões de injeção.

O processo de uma simulação de um reservatório é ilustrado na Fig. 1a (i.e., informar os controles dos poços, logo é executado o simulador e retorna repostas dos poços). No desenvolvimento de atividades de otimização da malha de drenagem, otimização de gerenciamento de poços e avaliação de

incertezas, é necessário um grande número de simulações, o que demanda elevado custo computacional e influencia o processo de tomada de decisão. Um exemplo é o Challenger Olympus, que otimiza o desenvolvimento do campo considerando um modelo de reservatório sintético [3].

Uma alternativa para que esses processos sejam mais práticos é o desenvolvimento de modelos proxy, baseados em dados do histórico de produção, substituindo o simulador com precisão adequada (Fig. 1b e 1c). Esta proposta permitirá entender a dinâmica do reservatório sem conhecer suas propriedades físicas, evitando custo computacional com simulações para tomar uma decisão sobre os controle dos poços.

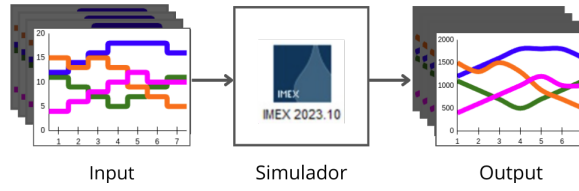
Os modelos proxy na engenharia de reservatório podem ser classificados como físicos, funcionais ou baseados em dados. Os modelos baseados em dados utilizam exclusivamente informações históricas de produção para análise e fornecimento de respostas, sem estabelecer uma formulação matemática explícita entre entradas e saídas. Dentro dessa categoria estão incluídos os modelos proxy baseados em redes neurais [4], que nos últimos anos têm sido aplicados para prever vazões e/ou pressões em reservatórios. Diversos modelos propostos na literatura são baseados em Redes Neurais Recorrentes (RNN) (ex. as rede tipo LSTM [5] e GRU [6]).

Essas abordagens geralmente só utilizam um único conjunto do histórico de produção ou apenas consideram dados históricos de uma única série temporal, desconsiderando a relação entre os múltiplos controles e não dando atenção a informações das mudanças dos controles ao longo do período considerado.

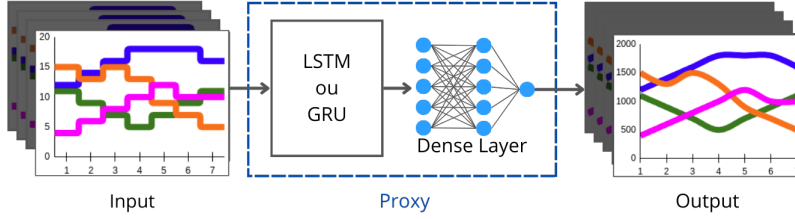
Diferentemente, neste trabalho, diversas séries temporais dos controles de poços injetores e produtores serão consideradas como entradas, gerando séries de respostas de BHP e/ou vazões nos injetores, assim como vazões de água e óleo nos poços produtores (Fig. 1b e 1c). Desse modo, propomos a utilização de controles mistos, nos quais os poços produtores são controlados por BHP, enquanto os poços injetores são controlados por vazão.

O restante do artigo é organizado como segue. A Seção II apresenta o referencial. A metodologia é detalhada na Seção III. Os resultados e discussões são descritos na Seção IV e, por fim, as conclusões são apresentadas na Seção V.

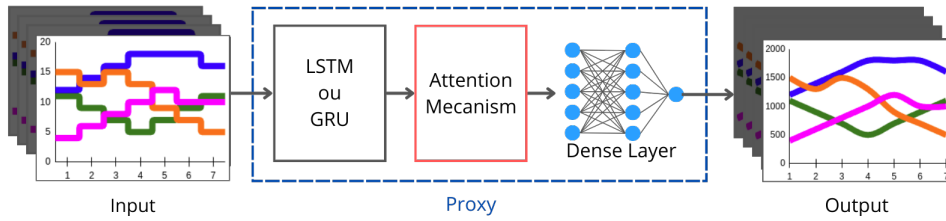
<sup>1</sup><https://github.com/johnattandouglas/RedesNeurais20241.git>



(a) Estratégia tradicional usando um simulador.



(b) Estratégia proposta com LSTM ou GRU.



(c) Estratégia proposta com LSTM ou GRU e mecanismo de atenção.

Fig. 1: Representação de substituição do simulador pelo modelo proxy. As entradas são os controles dos poços e as saídas são as produções de líquidos.

## II. METODOLOGIA

As RNN são estruturas de redes neurais artificiais específicas para tratar dados sequenciais como séries temporais ou linguagem natural. As RNN têm a capacidade de manter uma “memória” das observações anteriores mantendo e respeitando o contexto temporal do problema [7]. No nosso problema de previsão de vazão e/ou BHP dos poços, os modelos proxy baseados em RNN são relevantes pois a mudança de um controle não afeta apenas a resposta imediata, mas as respostas seguintes também.

### A. Long Short Term Memory (LSTM)

As LSTM são um tipo específico de RNN que aborda os problemas de vanishing e exploding gradients herdados da arquitetura original das redes recorrentes simples [7]. Para isso, implementam a ideia de portas de célula recorrente e estado oculto da célula. Existem três tipos de portas na LSTM: forget, input e output.

Na Fig. 2, os valores  $c$ ,  $h$ , e  $x$  representam o estado da célula (cell state), o estado oculto (hidden state) e a entrada (input), respectivamente, para cada passo de tempo  $t - 1$ . A porta forget avalia a importância das informações das células anteriores para a saída atual; a porta input decide a importância da entrada atual para a memória de longo prazo; e a porta output avalia a importância da entrada atual para a memória de curto prazo.

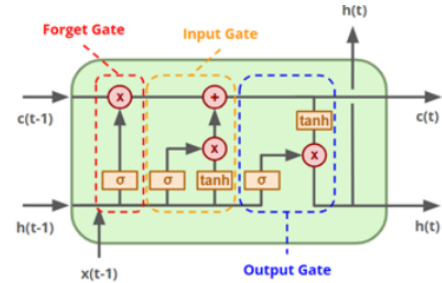


Fig. 2: Arquitetura LSTM.

A principal vantagem das portas é o controle do fluxo de memória na célula recorrente, determinando o quanto a célula considerará as informações do passado e do presente [8].

### B. Gate Recurrent Unit (GRU)

Semelhante as LSTM, as GRU também resolvem o problema de vanishing gradients das RNN, porém lidando com uma arquitetura mais simples de apenas dois portões: reset e update.

As GRU não têm uma memória interna exclusiva em cada célula, ou seja, não existe um estado de célula (Cho et al., 2014). A Fig. 3 exemplifica uma arquitetura GRU com uma célula. O portão reset é focado na memória a curto prazo, enquanto o update é focado na memória a longo prazo. As equações de ambos os portões são análogas as equações dos

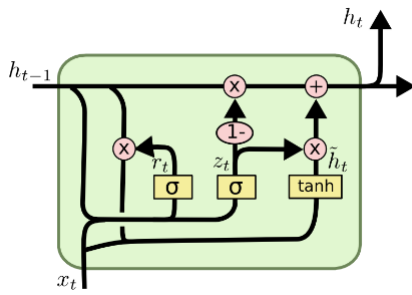


Fig. 3: Arquitetura GRU.

portões da LSTM. As saídas destes dois portões são agrupadas pelo estado escondido [9]. O estado escondido é construído por um processo de dois passos: definição de um candidato a estado escondido e combinação do estado escondido anterior com o candidato a estado escondido atual. O candidato a estado escondido é uma combinação da entrada atual, o estado escondido anterior e o portão reset, passando todos por uma função de ativação (Cho et al., 2014). Por fim, o candidato a estado escondido é combinado ao estado escondido anterior e ao portão update. A saída do portão determina o peso do candidato e do estado escondido anterior [9].

### C. Mecanismo de Atenção

Nas RNN convencionais, cada entrada corresponde a uma saída para o mesmo instante de tempo. Muitas vezes, queremos prever uma sequência de saída dado uma sequência de entrada de tamanho diferente. Esses modelos são conhecidos como sequence-to-sequence mapping. Para lidar com esses tipos de modelos, o modelo Encoder-Decoder foi desenvolvido, uma vez que ele recebe uma sequência como entrada e gera a próxima sequência provável como saída [10]. Embora esse modelo funcione bem para pequenas sequências, experimentos mostram que sua performance decai à medida que o tamanho da sequência cresce [9]. A alternativa para tratar essas limitações é o mecanismo de atenção.

O mecanismo de atenção é uma evolução do modelo Encoder-Decoder, desenvolvido para melhorar seu desempenho diante de longas sequências de entrada permitindo que o modelo dê “atenção” a diferentes partes da sequência de entrada ao gerar cada elemento da sequência de saída [11]. Especificamente, ele calcula uma ponderação para cada saída intermediária do codificador em relação ao passo de tempo atual do decodificador. Essas ponderações determinam a importância relativa de cada saída intermediária para a geração da saída atual. As etapas envolvem o cálculo dos pesos de atenção, determinando (i) a relevância de cada parte da entrada em relação ao estado atual do decodificador, (ii) o cálculo do contexto, que combina as partes mais relevantes das entradas ponderadas pelos pesos de atenção para formar o vetor de contexto, e (iii) a geração de saída, que utiliza o vetor de contexto para gerar a saída correspondente no passo de tempo atual.

Desse modo, ao incorporar o mecanismo de atenção, o modelo se torna capaz de focar em partes da entrada que são mais importantes para a geração de cada parte da saída, melhorando significativamente seu desempenho em tarefas com dependências temporais longas. Isso não só melhora a precisão das previsões, mas também melhora a interpretabilidade do modelo, possibilitando visualizar quais partes da entrada estão sendo priorizadas para cada saída gerada [11].

O mecanismo de atenção oferece uma vantagem em termos de interpretabilidade em comparação com outros modelos, frequentemente considerados “caixas-pretas” devido à sua incapacidade de explicar suas saídas. No contexto das redes neurais exploradas neste estudo, o LSTM destaca-se por resolver eficazmente o problema da explosão do gradiente. Contudo, durante o treinamento, o overfitting pode comprometer o desempenho no conjunto de teste, especialmente porque o LSTM fixa o vetor de entrada da rede neural em uma dimensão específica, o que é eficaz em espaços de baixa dimensão.

Para mitigar esses desafios e focar nos parâmetros influentes, introduzimos o mecanismo de atenção no LSTM, visando melhorar seu desempenho em sequências longas de entrada. O mecanismo de atenção permite que o codificador LSTM pondere de forma seletiva as saídas intermediárias em relação à sequência de entrada, facilitando o aprendizado desses inputs e sua associação com as saídas geradas pelo modelo [12]–[14]. O modelo GRU proposto com mecanismo de atenção (GRU-A) segue uma abordagem análoga ao LSTM-A.

## III. APLICAÇÃO

### A. Base de Dados

Com o intuito de produzir proxy capazes de substituir simuladores de reservatórios, foi utilizado um dataset baseado em um modelo de reservatório heterogêneo, chamado de Egg model [15]. O dataset possui vários canais de alta permeabilidade permitindo maior passo do fluido (Fig. 4).

A permeabilidade horizontal do reservatório varia de 81 e 7000 Millidarcy. A permeabilidade vertical é 10% da permeabilidade horizontal. A saturação inicial de água é 0.1, a porosidade é 0.2. O modelo possui 7 camadas, com 18.553 células. A pressão média do reservatório é 400 bar.

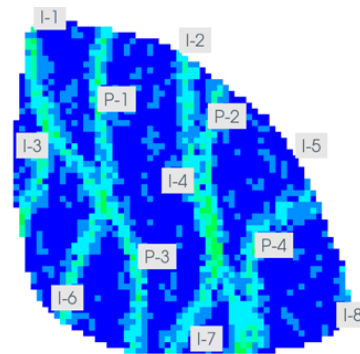
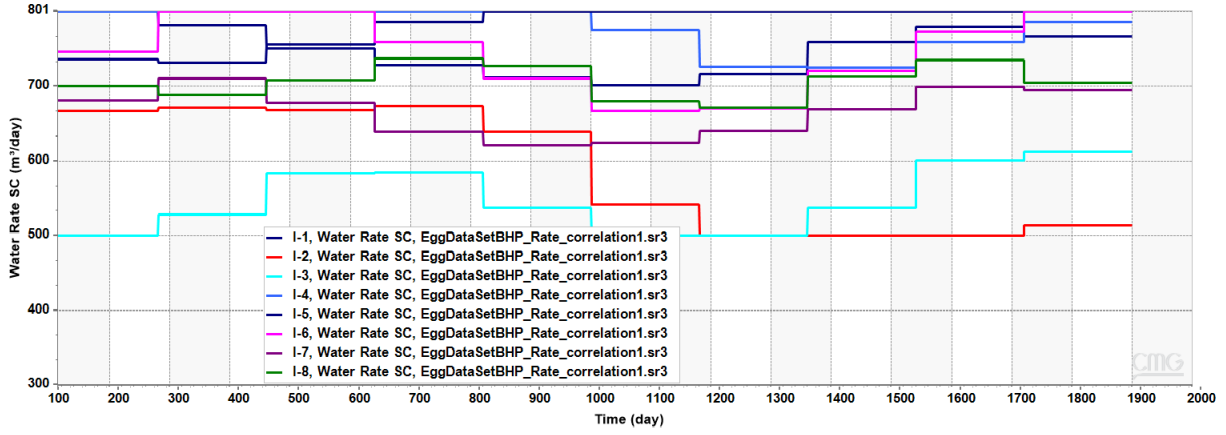
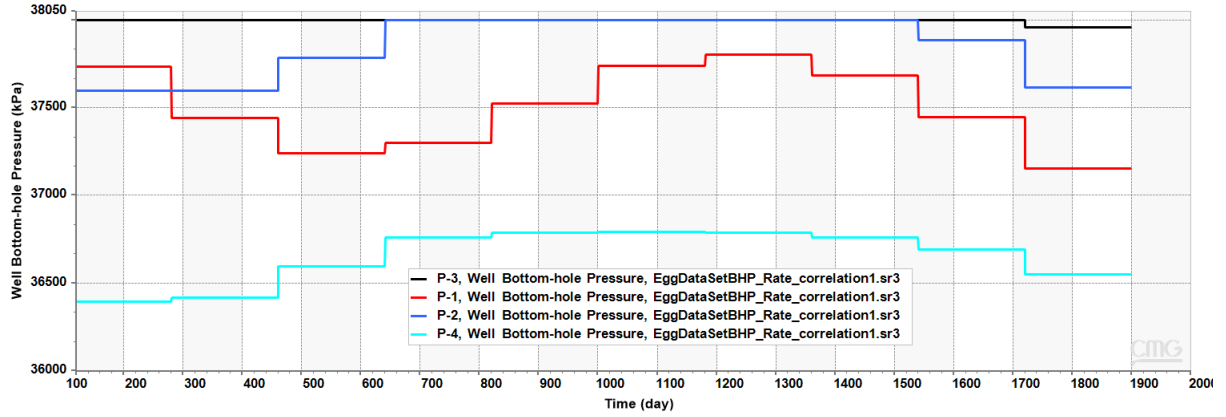


Fig. 4: Egg model reservoir.



(a) Vazão dos poços Injetores (I-1, I-2, I-3, I-4, I-5, I-6, I-7 e I-8)



(b) Pressão de Fundo de Poço (P-1, P-2, P-3 e P-4)

Fig. 5: Formato de entrada dos dados.

O reservatório possui 12 poços verticais, 4 produtores e 8 injetores. Nos campos de petróleo, os controles ou válvulas, de poços produtores e injetores não são alterados abruptamente, por essa razão foram gerados controles correlacionados para ambos os poços (Fig. 5). A partir dos controles correlacionados geramos um dataset com 1463 amostras.

O controle de cada poço produtor e injetor são alterados a cada 180 dias para o intervalo de [0, 1800] dias de simulação, ou seja, 10 ciclos de controles por poço. O intervalo de variação para os controles dos poços produtores e injetores são de 360 à 380 bar e de 500 à 800  $m^3$ /dia, respectivamente.

Na Fig. 5a exibimos os controle de vazão para os poços injetores e os controles de BHP dos produtores na Fig. 5b. As respostas são geradas após processo de simulação do modelo de reservatório. O simulador utilizado é o IMEX da Computer Modelling Group [16]. As saídas dos produtores são as respostas da produção de óleo e produção de água. No caso dos injetores a saída é a pressão dos poços injetores. As respostas dos poços são recuperados a cada 30 dias.

### B. Modelos utilizados

Os modelos utilizados como proxy para substituir o simulador foram LSTM e GRU (Fig. 1b); e LSTM-A e GRU-A

(Fig. 1c). Todos os experimentos foram realizados no Apuana<sup>2</sup>. Para todos os modelos foi definido um critério de parada de 500 épocas, com parada prematura configurada para 10 épocas de paciência [17].

### C. Métricas de Avaliação

A função objetivo utilizada no processo de treinamento é o Erro Médio Absoluto (MAE). Durante o teste, o otimizador Adam [18] buscou minimizar o Erro Quadrático Médio (RMSE) entre os dados observados e calculados. Em ambas as métricas,  $y_i$  representa os valores observados das vazões dos poços produtores e injetores,  $\hat{y}_i$  denota os valores calculados utilizando as estratégias propostas.

$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (1)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (2)$$

<sup>2</sup><https://apuana.cin.ufpe.br/>

#### D. Divisão dos dados

Os dados do histórico de produção para treinamento, validação e teste foram obtidos executando o simulador de reservatórios IMEX [16]. As respostas do simulador, isto é, as vazões de óleo e água produzida e a de água injetada ou pressão, são medidas em condições de superfície. Aqui, selecionamos aleatoriamente 70% do total das amostras para treinamento, 20% para validação e 10% para teste.

#### E. Otimização de Hiperparâmetros

Realizou-se uma busca em grade (grid search) para obter os hiperparâmetros ótimos, realizando uma combinação de valores predefinidos [19]. Os hiperparâmetros avaliados nas três arquiteturas foram: número de unidades por camada de 100 a 400 com variação de 50 unidades por busca. Tamanho de batch =  $2^i$ , com  $i = \overline{2:8}$ , taxa de aprendizagem variando ( $10^{-5}$ ,  $10^{-4}$ ,  $10^{-3}$ ) e função de ativação (linear, ReLU [20], Tanh). Os outros hiperparâmetros foram mantidos constantes. Os melhores hiperparâmetros para as estratégias LSTM e GRU após a busca em grade são mostrados na Tabela I.

A inicialização dos pesos na busca em grade foi aleatória. No LSTM-A e GRU-A foram considerados os hiperparâmetros encontrados para o LSTM e o GRU, respectivamente.

TABLE I: Hiperparâmetros ótimos após grid search.

Hiperparâmetro	LSTM	GRU
Número de Neurônios	400	350
Tamanho do Batch	4	4
Taxa de Aprendizagem	$10^{-3}$	$10^{-3}$
Função de Ativação	ReLU	Linear

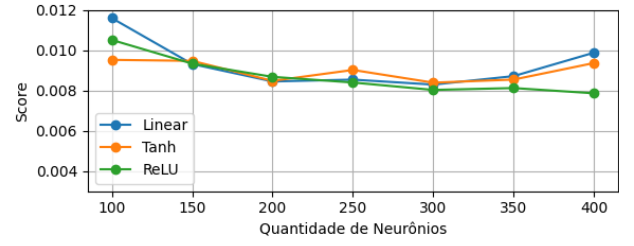
### IV. RESULTADOS E DISCUSSÕES

Percebeu-se que, para ambos os modelos base (LSTM e GRU), o tamanho ótimo de batch foi 4 e taxa de aprendizagem ótima foi  $10^{-3}$ . Então, a relação da função de ativação com o número de unidades dos modelos foi investigada. Na Fig. 6 é mostrado o score alcançados pelos modelos de diferentes tamanhos usando as três funções de ativação, considerando o tamanho de batch e taxa de aprendizagem ótimos encontrados.

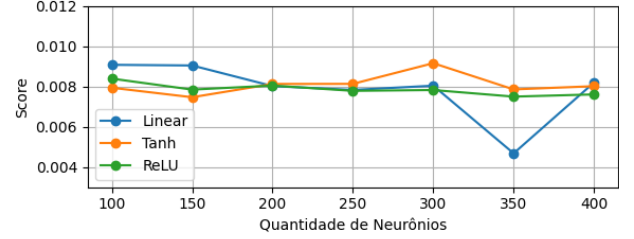
Com isso, observa-se que a medida que aumentamos o número de unidades nas arquiteturas dos modelos, a função de ativação ReLU reduz o erro no processo de treinamento.

A aleatoriedade no processo de divisão dos dados e na inicialização dos modelos pode impactar negativamente resultado obtido. Então, para uma análise mais consistente, cada modelo proxy foi executado 10 vezes para analisar sua distribuição de probabilidade, estes resultados são exibidos em gráficos de violino na Fig. 7.

O melhor resultado médio obtido foi utilizando a estratégia GRU-A com RMSE de 1.51%, mesmo com 40% menos hiperparâmetros treináveis em comparação ao LSTM e LSTM-A. Comparando os resultados entre GRU-A com LSTM e LSTM-A, obtivemos um ganho de 13%. Observa-se ainda que a adição do mecanismo de atenção, não modifica o número de parâmetros treináveis, mas aumenta o custo computacional.

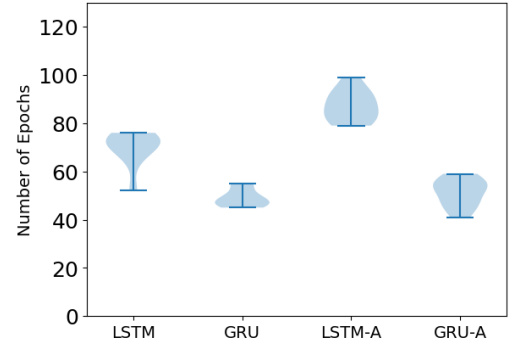


(a) LSTM

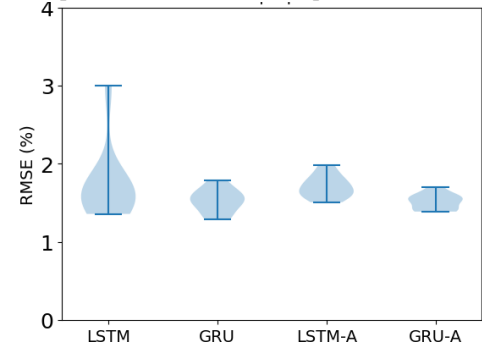


(b) GRU

Fig. 6: Relação da função de ativação com o número de unidades do modelo.



(a) Análise comparativo do RMSE médio para as diferentes estratégias.

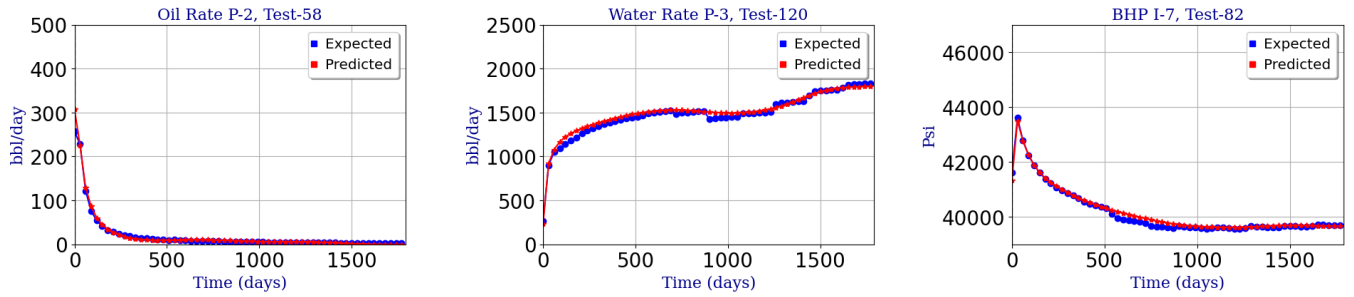


(b) Estudo comparativo do RMSE médio para as diferentes estratégias.

Fig. 7: Análise comparativa dos hiperparâmetros ótimos utilizando gráficos de violino.

Na Tabela II mostramos os valores médios de cada violino para o número de épocas e RMSE. Todas os modelos convergiram com parada prematura. A convergência antecipada do GRU e GRU-A em relação aos demais se dá pela quantidade de parâmetros treináveis.





(a) Óleo no poço produtor P-2 (teste #58). (b) Água no poço produtor P-3 (teste #120). (c) BHP no poço injetor I-7 (teste #82).

Fig. 8: Previsão no conjunto de teste.

TABLE II: Comparação das estratégias propostas.

	Parâmetros treináveis	Nº de Épocas	RMSE %
LSTM	667216	70	1.72
LSTM-A	667216	87	1.71
GRU	387816	62	1.53
GRU-A	387816	74	1.51

Nas Fig.s 8a, 8b e 8c exibimos a previsão da produção de óleo no produtor P-1, produção de água no produtor P-3 e a pressão de fundo de poço do poço injetor I-2, respectivamente, todas elas obtidas utilizando a estratégia GRU-A.

## V. CONCLUSÕES

Neste trabalho foi apresentado um modelo proxy baseado em redes neurais para aprender a dinâmica do simulador de reservatório para prever as respostas de vazão de óleo e água nos produtores e pressão dos poços injetores, baseado em séries de controles de pressão e vazão nos poços produtores e injetores, respectivamente.

A estratégia de busca em grade demonstrou ser eficiente na determinação dos hiperparâmetros ótimos das arquiteturas propostas, revelando que a função ReLU é mais eficaz quando combinada com um maior número de unidades.

Os modelos proxy de LSTM e GRU exibem maior variabilidade nos resultados. Entretanto, ao introduzir o mecanismo de atenção em ambas as estratégias, observa-se uma redução na variabilidade dos resultados, embora isso leve a um aumento no tempo de processamento necessário para obter tais resultados.

Utilizando a estratégia GRU-A, alcançamos o melhor resultado médio com um RMSE de 1.51%, mesmo com 40% menos hiperparâmetros treináveis em comparação ao LSTM e LSTM-A, representando um ganho de desempenho de 13%. O GRU-A pode ser efetivamente empregado como modelo proxy para substituir um simulador e prever as respostas dos poços produtores e injetores.

## ACKNOWLEDGMENT

The preferred spelling of the word “acknowledgment” in America is without an “e” after the “g”. Avoid the stilted expression “one of us (R. B. G.) thanks ...”. Instead, try “R. B. G. thanks...”. Put sponsor acknowledgments in the unnumbered footnote on the first page.

## REFERENCES

- [1] Z. Chen, *Reservoir Simulation: Mathematical Techniques in Oil Recovery*. SIAM, 2007.
- [2] K. Aziz and A. Settari, *Petroleum Reservoir Simulation*. Applied Science Publishers, 1979.
- [3] R. G. Fonseca, “Description of olympus reservoir model for optimization challenge,” in *Integrated Systems Approach to Petroleum Production*, Netherlands, 2017.
- [4] W. S. McCulloch and W. Pitts, “A logical calculus of ideas immanent in nervous activity,” *Bulletin of Mathematical Biophysics*, vol. 5, no. 4, pp. 115–133, 1943.
- [5] D. Zhang, Q. Peng, J. Lin, D. Wang, X. Liu, and J. Zhuang, “Simulating reservoir operation using a recurrent neural network algorithm,” *Water*, vol. 11, no. 4, pp. 865–882, 2019.
- [6] Y. Jiang, W. Shen, H. Zhang, K. Zhang, J. Wang, and L. Zhang, “An interpretable recurrent neural network for waterflooding reservoir flow disequilibrium analysis,” *Water*, vol. 15, pp. 623–639, 2023.
- [7] J. Brownlee, *Deep learning for time series forecasting*. Machine Learning Mastery, 2018.
- [8] A. Graves and A. Graves, “Long short-term memory,” in *Supervised Sequence Labeling with Recurrent Neural Network*, 2012, pp. 37–45.
- [9] K. Cho, B. van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, “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)*, A. Moschitti, B. Pang, and W. Daelemans, Eds. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 1724–1734. [Online]. Available: <https://aclanthology.org/D14-1179>
- [10] J. C. Ye and W. K. Sung, “Understanding geometry of encoder-decoder cnns,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 7064–7073.
- [11] K. Aitken, V. Ramasesh, Y. Cao, and N. Maheswaranathan, “Understanding how encoder-decoder architectures attend,” *Advances in Neural Information Processing Systems*, vol. 34, pp. 22 184–22 195, 2021.
- [12] Q. Kang, E. J. Chen, Z.-C. Li, H.-B. Luo, and Y. Liu, “Attention-based lstm predictive model for the attitude and position of shield machine in tunneling,” *Underground Space*, vol. 13, pp. 335–350, 2023.
- [13] Z. Liu, W. Zhou, and H. Li, “Ab-lstm: Attention-based bidirectional lstm model for scene text detection,” *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, vol. 15, no. 4, pp. 1–23, 2019.
- [14] T. Luo, X. Cao, J. Li, K. Dong, R. Zhang, and X. Wei, “Multi-task prediction model based on convlstm and encoder-decoder,” *Intelligent Data Analysis*, vol. 25, no. 2, pp. 359–382, 2021.
- [15] J. Jansen, R. Fonseca, S. Kahrobaei, M. Siraj, G. Essen, van, and P. Hof, Van den, “The egg model - a geological ensemble for reservoir simulation,” *Geoscience Data Journal*, vol. 1, no. 2, pp. 192–195, 2014.
- [16] Computer Modeling Group LTD., *IMEX. Em User’s Guide*, Calgary, Canada, 2022.
- [17] Y. Yao, L. Rosasco, and A. Caponnetto, “On early stopping in gradient descent learning,” *Constructive Approximation*, vol. 26, no. 2, pp. 289–315, 2007. [Online]. Available: <https://doi.org/10.1007/s00365-006-0663-2>

- [18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2017.
- [19] S. M. LaValle, M. S. Branicky, and S. R. Lindemann, "On the relationship between classical grid search and probabilistic roadmaps," *The International Journal of Robotics Research*, vol. 23, no. 7-8, pp. 673–692, 2004.
- [20] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *International Conference on Machine Learning*, 2010, pp. 807–814.