



**Taisa Dornelas Abbas Calvette**

**Aplicação de técnicas de *machine learning* e *data driven*  
em poços inteligentes de petróleo**

**Dissertação de Mestrado**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-Graduação em Engenharia Elétrica do Departamento de Engenharia Elétrica da PUC-Rio.

Orientador: Prof. Marco Aurélio C. Pacheco

Coorientador: Prof. Ana Carolina Abreu

Rio de Janeiro  
Agosto de 2019



**Taisa Dornelas Abbas Calvette**

**Aplicação de técnicas de *machine learning* e *data driven*  
em poços inteligentes de petróleo**

Dissertação apresentada como requisito parcial para obtenção do grau de Mestre pelo Programa de Pós-graduação em Engenharia Elétrica da PUC-Rio. Aprovada pela Comissão Examinadora abaixo.

**Prof. Marco Aurélio C. Pacheco**

Orientador

Departamento de Engenharia Elétrica – PUC-Rio

**Prof. Ana Carolina Abreu**

Coorientadora

Departamento de Engenharia Elétrica – PUC-Rio

**Prof. Marco Antonio Guimaraes Dias**

Departamento de Engenharia Elétrica – PUC-Rio

**Prof. Manoela Rabello Kohler**

Departamento de Engenharia Elétrica – PUC-Rio

**Prof. Leonardo Alfredo Forero Mendonza**

Universidade Estadual do Rio de Janeiro – UERJ

Rio de Janeiro, 06 de agosto de 2019

Todos os direitos reservados. É proibida a reprodução total ou parcial do trabalho sem autorização da universidade, da autora e do orientador.

### **Taisa Dornelas Abbas Calvette**

Graduada em Ciências Econômicas no IBMEC, possui pós-graduação em Gestão Ambiental e Geoprocessamento pela Uni-BH e é Mestre em Engenharia Urbana e Ambiental pela PUC -Rio e pela Technische Universität Braunschweig. Participou do Programa de estágio internacional no Inter-American Development Bank em Washington DC e já atuou como consultora na Petrobras e na Petranspor.

#### Ficha Catalográfica

Calvette, Taisa Dornelas Abbas

Aplicação de técnicas de *machine learning* e *data driven* em poços inteligentes de petróleo / Taisa Dornelas Abbas Calvette ; orientador: Marco Aurélio C. Pacheco ; co-orientador: Ana Carolina Abreu. – 2019.

68 f. : il. color. ; 30 cm

Dissertação (mestrado)–Pontifícia Universidade Católica do Rio de Janeiro, Departamento de Engenharia Elétrica, 2019.

Inclui bibliografia

1. Engenharia Elétrica – Teses. 2. Machine learning. 3. Proxy. 4. Smart proxy. 5. Deep smart proxy. 6. Poços inteligentes. I. Pacheco, Marco Aurélio Cavalcanti. II. Abreu, Ana Carolina Alves. III. Pontifícia Universidade Católica do Rio de Janeiro. Departamento de Engenharia Elétrica. IV. Título.

CDD: 621.3

Para Luigi, Olivia e Tetê.

## Agradecimentos

Agradeço primeiramente à Deus e a São Francisco de Assis por me guiarem em todas as escolhas da minha vida.

Ao meu esposo Luigi, pelo apoio nos momentos mais difíceis e cansativos e pelo amor incondicional todos os dias.

À minha filha Olivia que me ensina a cada dia a ser uma pessoa melhor.

À minha Golden retriever Tetê pelo companheirismo e amor incondicional.

Aos meus pais, pela educação, amor e carinho. Vocês me proporcionaram não somente a oportunidade de obter mais este diploma, como todas as ferramentas necessárias para ser feliz na vida.

Aos grandes professores da PUC-Rio, em especial ao meu orientador Marco Aurélio, por ser uma grande inspiração como professor, profissional e pessoa. Obrigada pelas importantes contribuições e ajuda sempre que precisei.

À querida Prof. Ana Carolina que carinhosamente me orientou e ajudou com a dissertação. Você é uma fantástica pessoa e profissional.

Ao amigo Marco Antonio Dias por todo apoio necessário durante todo o mestrado e pela amizade durante todos estes anos.

Aos colegas do ICA que de alguma forma me estimularam e me ajudaram, em especial Leonardo, Manuela, Cristian, Allan e Juliana.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) – Código de Financiamento 001.

## Resumo

Calvette, Taisa Dornelas Abbas; Pacheco, Marco Aurélio C. (Orientador); Abreu, Ana Carolina (Coorientador). **Aplicação de técnicas de *machine learning* e *data driven* em poços inteligentes de petróleo/ Taisa Dornelas Abbas Calvette**. Rio de Janeiro, 2019. 68p. Dissertação de Mestrado. Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

Realizar uma estimativa confiável na produção de petróleo é um dos grandes desafios na indústria de óleo e gás e é uma parte crítica no planejamento e na tomada de decisão das petrolíferas. Neste contexto, este trabalho visa explorar as vantagens e desempenho dos algoritmos de *machine learning* para realizar a previsão de produção de petróleo, gás e água a partir das informações de controle de poços inteligentes e usando a metodologia de *data driven*. Para tanto, foram usadas duas bases de dados com séries históricas de produção de petróleo, gás e água. A primeira base foi gerada sinteticamente (através de simulação de reservatórios) e consiste na produção média mensal e configuração de 3 válvulas de um poço injetor, ao longo de um período de 10 anos. A segunda base usa dados reais de produção (observados) fornecidos pelo estado da Dakota do Sul nos Estados Unidos. Esta base consiste na média diária de produção e o estado geral (ativo ou não produzindo) de diversos poços produtores de petróleo no período compreendido de 1950 a 2018. Com o intuito de testar a metodologia, foram realizados diversos experimentos combinando o treinamento da *proxy* com algoritmos de Redes Neurais Artificiais (*Multilayer Perceptron*) e *deep learning* com redes neurais recorrentes (redes neurais recorrentes simples, *long short-term memory*, *Gated Recurrent Units*), chamados de *smart proxy* e *deep smart proxy* respectivamente. Os resultados encontrados mostraram que o modelo *deep smart proxy* se mostrou bastante promissor. Utilizando uma rede *Gated Recurrent Units* com camadas bidirecionais (GRUB), foi possível obter uma redução no erro RMSE de 66% e no erro MAE de 79% quando comparados aos modelos *smart proxy* com Redes Neurais Artificiais. Verificou-se que nos modelos *deep smart proxy*, o uso de camadas bidirecionais gerou uma significativa melhora na previsão e redução do erro, tanto nos testes que utilizaram dados de produção simulados (caso sintético)

quanto nos testes que utilizaram dados de produção observados (caso real), proporcionando uma variação de até 75% no RMSE e 85% no MAE. O erro RMSE normalizado na rede GRUB foi de 0,53% nos dados observados e 0,65% nos dados sintéticos. Os modelos de *deep smart proxy* obtiveram desempenhos muito semelhantes, principalmente ao comparar o desempenho das redes do tipo LSTMB e GRUB. Estas redes foram aplicadas em ambos os casos sintético e real de produção e superaram, em todos os casos, os resultados obtidos com o modelo de *smart proxy* com MLP.

## **Palavras-chave**

*Machine learning; proxy; smart proxy; deep smart proxy; poços inteligentes; produção de petróleo.*

## Abstract

Calvette, Taisa Dornelas Abbas; Pacheco, Marco Aurélio C. (Advisor); Abreu, Ana Carolina (Co-Advisor). **Application of machine learning and data driven techniques to smart oil wells/ Taisa Dornelas Abbas Calvette**. Rio de Janeiro, 2019. 68p. Dissertação de Mestrado - Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro.

A reliable forecast for oil production represents one of the biggest challenges in the oil and gas industry and contributes to the planning and decision making of oil companies. Because of that, this work uses intelligent well valves settings and *data driven* methodology to explore the advantages and the performance of machine learning algorithms in the forecasting of oil, gas and water production. In order to do so, two database containing historical data series of oil, gas and water production were used. The first was generated synthetically (through reservoir simulation) and consisted of the average monthly production of an injection well over a period of 10 years, as well as the configuration of 3 of its valves. The second database used the production data provided by the state of South Dakota, located in the United States, and consisted of the daily production average and the overall well status (active or not producing) from several oil producing wells in a period ranging from 1950 to 2018. In order to test the methodology, several experiments were performed combining proxy with Artificial Neural Network Algorithms (Multilayer Perceptron) and deep learning recurrent neural networks (Simple Recurrent Neural Networks, long short-term memory, Gated Recurrent Units), which were named smart proxy and deep smart proxy, respectively. The results showed that the deep smart proxy model was very promising. Using the Gated Recurrent Units network with bi-directional layers (GRUB), a reduction of 66% in the RMSE error and 79% in the MAE error was obtained when compared to smart proxy models with Artificial Neural Networks. The deep smart proxy models with bidirectional layers generated a significant improvement in prediction and error reduction in both databases tests ( i.e. tests with simulated production data (synthetic case) and with the observed production data (real case), resulting in a variation of up to 75% in RMSE and 85% in MAE). The normalized RMSE error in the GRUB network was of 0.53% in the observed

database and 0.65% in the synthetic database. It is important to notice that the Deep smart proxy models achieved very similar performances when comparing the LSTMB and GRUB network in both databases (synthetic and real production), surpassing in all cases the results obtained with the MLP smart proxy model.

## **Keywords**

Machine learning; proxy; smart proxy; deep smart proxy; smart wells; oil production.

# Sumário

<b>1. Introdução</b>	<b>14</b>
1.1. Motivação	14
1.2. Objetivo do trabalho	16
1.3. Organização do trabalho	19
<b>2. Poços inteligentes com aplicação de <i>data driven</i></b>	<b>20</b>
2.1. Poços inteligentes	20
2.2. <i>Data driven</i>	22
<b>3. Redes Neurais Artificiais e <i>deep learning</i></b>	<b>27</b>
3.1. Redes <i>Multilayer Perceptron</i>	28
3.2. Redes <i>deep learning</i>	30
3.2.1. Redes Neurais Recorrentes (RNR)	31
3.2.2. Redes <i>long short-term memory</i> (LSTM)	34
3.2.3. Redes <i>Gated Recurrent Unit</i> (GRU)	38
3.2.4. Camadas bidirecionais	40
<b>4. <i>Smart proxy</i> e <i>deep smart proxy</i></b>	<b>42</b>
<b>5. Previsão com dados simulados - modelos <i>smart proxy</i> e <i>deep smart proxy</i></b>	<b>45</b>
5.1. Base de dados	45
5.2. <i>Smart proxy</i> – Testes e resultados	46
5.3. <i>Deep smart proxy</i> – Testes e resultados	49
<b>6. Previsão com dados observados</b>	<b>53</b>
6.1. Base de dados	53
6.2. Mineração de dados ( <i>Data Mining</i> )	53
6.3. Testes e resultados	55
<b>7. Conclusão</b>	<b>61</b>
<b>8. Sugestão para trabalhos futuros</b>	<b>63</b>
<b>9. Referências bibliográficas</b>	<b>64</b>

## Lista de tabelas

Tabela 1 – Configuração dos Nós do Cluster do IntelDevCloud	46
Tabela 2 – Parâmetros das Redes Neurais Artificiais (MLP)	47
Tabela 3 – Resultados dos testes <i>smart proxy</i> usando Redes Neurais <i>Multilayer Perceptron</i>	48
Tabela 4 – Resultados dos testes <i>deep smart proxy</i> com a Rede Neural Recorrente Simples (RNNS)	50
Tabela 5 – Resultados dos testes <i>deep smart proxy</i> com a Rede <i>long short-term memory</i> (LSTM)	51
Tabela 6 – Resultados dos testes <i>deep smart proxy</i> com a Rede <i>Gated Recurrent Unit</i> (GRU)	51
Tabela 7 – Situação do poço e variável binária correspondente	54
Tabela 8 – Resultados dos testes com dados observados usando Redes Neurais Multilayer Perceptron (MLP)	56
Tabela 9 – Resultados dos testes com dados reais usando a Rede Neural Recorrente Simples (RNNS)	57
Tabela 10 – Resultados dos testes com dados reais usando a Rede <i>long short-term memory</i> (LSTM)	58
Tabela 11 – Resultados dos testes com dados reais usando a Rede <i>Gated Recurrent Unit</i> (GRU)	59

## Lista de figuras

Figura 1 – Representação do funcionamento do <i>smart well</i>	20
Figura 2 – Representação <i>data driven</i>	22
Figura 3 – Representação do neurônio artificial	29
Figura 4 – Representação da Rede Neural <i>Multilayer Perceptron (Feedforward)</i>	29
Figura 5 – Comparativo da estrutura da RNR e RNA <i>feedforward</i>	31
Figura 6 – Arquitetura de uma RNR padrão com 3 camadas ocultas	32
Figura 7 – Estrutura da unidade da RNR (RNN)	33
Figura 8 – Estrutura da unidade da rede LSTM	34
Figura 9 – Portão de esquecimento ( <i>Forget</i> )	35
Figura 10 – Portão de entrada ( <i>Input</i> ) e de candidatura ( <i>Candidate</i> )	36
Figura 11 – Atualização do estado celular	37
Figura 12 – Portão de saída ( <i>Output</i> )	37
Figura 13 – Estrutura da unidade da rede GRU	38
Figura 14 – Arquitetura da LSTM bidirecional	40
Figura 15 – Modelo do reservatório sintético	44
Figura 16 – Arquitetura da rede MLP no modelo <i>smart proxy</i>	47

## Lista de siglas

CMG – *Computer Modelling Group*

CRM – *Capacitance–Resistance Models*

DCA – *Decline Curve Analysis*

GAN - *Generative Adversary Networks*

GRU – *Gated Recurrent Units*

LSTM – *Long short-term memory*

MAE – *Media Absoluta do Erro*

MAPE - *Média Absoluta do Erro Percentual*

MLP – *Multilayer Perceptron*

MRC – *Modelos de Resistência a Capacitância*

MSE – *Erro Quadrático Médio*

PCI – *Poço com Completação Inteligente*

RMSE – *Raiz do Erro Médio Quadrático*

RNA – *Redes Neurais Artificiais*

RNNS – *Rede Neural Recorrente Simples*

RNR – *Rede Neural Recorrente*

SPC – *Subspace Predictive Controler*

TANH – *Tangente Hiperbólica*

VBA – *Visual Basic for Applications*

VPL – *Valor Presente Líquido*

# 1

## Introdução

### 1.1

#### Motivação

A obtenção de uma estimativa confiável da produção de petróleo vem sendo um grande desafio na indústria de óleo e gás por ser uma parte crítica no planejamento e na tomada de decisão. De forma geral, as projeções de longo prazo afetam as estratégias das petrolíferas, bem como a seleção do portfólio de investimentos, contratos de vendas, promessas a *stakeholders* (Rajvanshi et al., 2012), além de ter relação direta com o fluxo de caixa futuro esperado nas decisões e no plano de negócios (Jaja et al., 2006).

Por se tratar de um campo de petróleo específico, a estimativa de produção de um poço afeta o processo de otimização, intervenção e os planos de gestão dos reservatórios (Rajvanshi et al., 2012). Segundo Jaja et al. (2006), apenas 45% das estimativas de produção de petróleo resultam em números próximos da realidade. Projeções muito otimistas costumam resultar em produções subdimensionadas, gerando inúmeras consequências. Por outro lado, projeções muito pessimistas resultam em uma subvalorização dos ativos (Rajvanshi et al., 2012).

Existem muitas incertezas que afetam a previsibilidade da produção. Estas envolvem desde a coleta adequada dos dados até a análise final do processo de previsão em si (Monteiro et al., 2017). Rajvanshi et al. (2012) cita algumas das principais razões pelas quais a produção estimada difere da real, dentre elas, o declínio mais acentuado da produção real frente ao previsto devido às hipóteses idealizadas sobre o reservatório (a simulação do poço tende a colocá-lo em operação durante um período maior) e erros na estimativa do pico e duração total de produção, resultando em uma produção menor de petróleo e maior de água.

Os métodos estatísticos clássicos mais comuns baseados no histórico de produção para previsão (como a análise da curva de declínio ou *Decline Curve Analysis*; análise de tendências e extrapolações; modelo *Black Oil* de simulação; dentre outros) são muito imprecisos pelo fato de serem fundamentados por uma

interpretação subjetiva dos dados que não fornece uma relação clara sobre o futuro (Weiss et al., 2002), o que mostra a necessidade de se ter uma metodologia capaz de realizar uma previsão com menor erro.

Outro importante fator que deve ser levado em consideração é que, como o petróleo é um recurso finito, é cada vez mais caro produzir e encontrar reservatórios, o que leva a uma taxa de recuperação de reservatório em torno de 35% (Gao et al., 2007).

O simulador de reservatório é uma ferramenta bastante utilizada para análise e previsão de produção devido a sua precisão. O simulador utiliza o método de diferenças finitas que busca modelar o reservatório por meio de um conjunto de variáveis e equações que descrevem todo o processo de escoamento de fluídos. No entanto, apesar de ser bastante preciso, Kohler (2013) ressalta que o custo operacional em termos de esforço computacional de um simulador se torna muito alto e, em grande parte, inviável para fazer várias análises (ex: otimização de localização de poços, análise de incertezas de reservatórios, dentre outras).

Como forma de otimizar o esforço e o tempo computacional, os modelos *proxy* surgem como alternativa para substituir de forma parcial os simuladores de reservatórios. Uma vez que os modelos *proxy* são em grande parte baseados em métodos matemáticos e estatísticos, eles incorrem em problemas comuns da inferência estatística, tais como: a correlação e causalidade entre variáveis, uma maior margem de erro em relação ao simulador e restrição na variação dos valores estimados (valores mínimo e máximo) de acordo com os valores da amostragem (Mohaghegh, 2015).

Com os avanços no processamento computacional e inteligência artificial, principalmente nos algoritmos de *machine learning*, surgiram os modelos chamados de *smart proxy*. Estes, por utilizarem Redes Neurais Artificiais, conseguem aprender o comportamento e reconhecimento de padrões, reproduzindo, assim, as condições do simulador de reservatório de forma mais rápida e com menor erro (Mohaghegh, 2015).

Desta forma, é possível dizer que, devido à inteligência artificial, bem como o desenvolvimento de novas técnicas e metodologias na simulação e avaliação da grande quantidade de dados disponíveis, os modelos empíricos dos reservatórios de óleo e gás vêm sofrendo grandes transformações (Holdaway, 2014). Gao et al.

(2007) explica que estas inovações fazem com que a meta da taxa de recuperação do reservatório das petrolíferas se torne 60%.

Dentre as principais inovações estão os sistemas de poços inteligentes (*smart wells*), através dos quais é possível o monitoramento remoto do fundo do poço, permitindo, por exemplo, o controle do poço a partir da superfície (Goh et al., 2008). Os poços inteligentes geram inúmeros benefícios para a gestão - principalmente para a otimização operacional na indústria de óleo e gás - e fornecem dados e insumos para uma melhor interpretação das características do reservatório. Os dados fornecidos pelos poços inteligentes, por sua vez, alimentam os simuladores de reservatórios usados nos modelos *smart proxy*. Desta forma, o simulador utiliza *smart wells* para gerar dados que são usados para treinar algoritmos de *machine learning* (como as Redes Neurais Artificiais e *deep learning*, encontrado em redes neurais recorrentes), juntamente com a metodologia de *data driven*, para fornecer uma melhor compreensão sobre o andamento das operações, auxiliando na otimização da produção e fornecendo insumos para a tomada de decisões.

## 1.2

### Objetivo do trabalho

O sistema de poços inteligentes é uma tecnologia emergente na qual são fornecidas oportunidades de melhorias na produção de petróleo e gás através de um sistema de injeção e produção mais equilibrado e ordenado (Ranjith et al., 2017). Utilizando sensores e válvulas, os poços inteligentes permitem o controle do fundo do poço pela superfície, gerando inúmeros benefícios principalmente devido ao sistema de válvulas, que contribui para a otimização da produção.

De forma geral, os dados dos poços inteligentes são fornecidos em tempo real e, por meio deles, é possível realizar o monitoramento, controle, otimizações, simulações e análises *data driven* (guiada por dados) do reservatório. Diante disso, os dados fornecidos pelo sistema de poços inteligentes são de suma importância, principalmente no que tange ao entendimento das características e gestão do reservatório. Através das simulações, é possível realizar a previsão de produção de petróleo, gás e água.

Considerando as alternativas existentes para a previsão da curva de produção, há, naturalmente, um *tradeoff* entre velocidade de resposta, custo, margem de erro e esforço computacional. As realizações de simulações de reservatórios possuem uma baixa margem de erro, mas exigem um alto esforço computacional, muito tempo e custo elevado, o que, diversas vezes, as torna inviáveis (Kohler, 2013). É neste contexto que são utilizados aproximadores ou modelos *proxy*, que, visando otimizar o tempo e reduzir o custo, substituem a necessidade da simulação integrada de reservatório e do sistema de produção (parcial ou total), através do uso de técnicas estatísticas, matemáticas e até mesmo de algoritmos de *machine learning*. Este último possui uma capacidade de aprendizado que o difere dos modelos convencionais de *proxy* (que em grande parte usam métodos estatísticos). Neste trabalho mais adiante, seu uso acarretará no chamado *smart proxy*, conforme sugerido por Mohaghegh (2015) e He et al. (2016).

O uso de *proxy*, especialmente com redes neurais (chamado neste trabalho de *smart proxy*), é amplamente empregado na engenharia de petróleo para fazer a modelagem de reservatórios devido a sua habilidade de processamento de informações em paralelo e distribuído com significativa redução do tempo computacional (Talavela et al., 2010).

Como exemplo, Talavela et al. (2010) usou a *proxy* com algoritmos de redes neurais *Multilayer Perceptron* com retropropagação (*Backpropagation*) com dados poços inteligentes para gerar uma aproximação do modelo de reservatório e o uso de *Reinforcement Learning* para encontrar o controle ótimo de válvulas e previsão de produção. Kohler (2013) também usou *proxy* e redes neurais para aproximar o comportamento de uma simulação integrada entre reservatório e sistema de superfície, sendo possível alterar a localização, geometria e tipo de poço.

Apesar do uso de inteligência artificial já fazer parte do setor de petróleo desde a década de 70, a implantação de novas técnicas de *machine learning*, acarretando em previsões cada vez mais precisas, está impulsionando a fronteira do conhecimento no setor de óleo e gás e, por consequência, está assumindo um papel proeminente no processo de exploração e produção (Maucec e Garni, 2019).

Nos últimos anos, os algoritmos de aprendizado profundo (*deep learning*) tem ganhado destaque em diversas aplicações, sendo as principais na identificação de imagens, previsões e reconhecimento da fala. De uma maneira geral, a estrutura

multicamadas e o processamento do *deep learning* fornecem a capacidade de resolver problemas complexos e encontrar padrões de forma inovadora. Na produção de petróleo e gás, estes algoritmos vêm sendo aplicados em estudos de geologia, otimização de produção, simulações de reservatórios (Zhang et. al., 2019) e previsão com base em séries históricas de produção (Sun et. al., 2018).

Dentre as técnicas usadas na previsão de produção, a Rede Neural Recorrente (RNR) é um dos algoritmos de *deep learning* mais usados devido sua capacidade de memorização no aprendizado. Algumas variações das RNRs, como a rede *long short-term memory* (LSTM) e *Gated Recurrent Units* (GRU), foram desenvolvidas e melhoraram significativamente o desempenho das RNRs devido a sua capacidade de aprendizado de dependências de longo prazo (Goodfellow et. al, 2016). Nesta linha, outra forma de melhoria das RNRs foi a aplicação de camadas bidirecionais tanto para a RNR convencional, quanto para a LSTM e GRU, cujas sequências de entrada anteriores e posteriores são usadas para melhor desempenho do processo de aprendizagem.

O principal objetivo deste trabalho é explorar as vantagens e desempenho dos algoritmos de *machine learning* para realizar a previsão de produção de óleo, gás e água, a partir dos dados sintéticos de poços inteligentes e usando a metodologia de *data driven*. Para tal, foram realizados diversos experimentos usando os aproximadores com o uso de algoritmos de Redes Neurais Artificiais (*Multilayer Perceptron*) e *deep learning* do tipo redes neurais recorrentes (redes neurais recorrentes simples, LSTM, GRU), chamados de *smart proxy* (He et al., 2016) e *deep smart proxy* respectivamente.

A pesquisa irá investigar o potencial de integração entre o uso de aproximadores (*proxy*), *data driven* e algoritmos de *machine learning* em especial *deep learning*, no setor de petróleo e gás, na previsão e otimização de produção. Além disso, com o intuito de testar a metodologia proposta, foram realizados os mesmos testes em uma base com um conjunto de dados observados (reais) de produção de diferentes poços e reservatórios. Vale ressaltar que não foram encontradas publicações de testes com a bases de dados observados e o uso de *deep learning*, como proposto neste trabalho.

### 1.3

#### Organização do trabalho

O estudo está dividido em oito capítulos, índice de tabelas, índice de figuras e referências bibliográficas. O capítulo 1 é a introdução do trabalho e apresenta 3 itens, sendo eles: motivação, objetivo do trabalho e organização do trabalho.

O capítulo 2 fornece uma explicação sobre o funcionamento dos poços inteligentes e a aplicação da metodologia de *data driven* no setor de petróleo.

O capítulo 3 faz uma breve explicação sobre o funcionamento das redes neurais Multilayer Perceptron (MLP) e redes de aprendizado profundo (*deep learning*) do tipo Redes Neurais Recorrentes.

O capítulo 4 aborda as principais diferenças e características dos modelos *proxy*, *smart proxy* e *deep smart proxy*.

O capítulo 5 detalha a base de dados sintéticos usada e apresenta o desempenho dos testes e resultados dos modelos *smart proxy* e *deep smart proxy* para a previsão de produção de óleo, gás e água.

O capítulo 6 descreve o teste a partir da base de dados observados com séries históricas reais de produção. Para tanto, foi detalhada toda a base de dados, descrito todo o processo de mineração de dados e realizado um comparativo do desempenho dos algoritmos de *machine learning* utilizados anteriormente para a previsão de produção de óleo, gás e água. O capítulo 7 apresenta a conclusão e considerações finais. O capítulo 8 apresenta as sugestões para futuros estudos.

## 2

### Poços inteligentes com aplicação *data driven*

#### 2.1

##### Poços inteligentes

Poços inteligentes estão sendo implementados cada vez mais na exploração de petróleo visando maximizar a produção de óleo e gás. Um poço inteligente, também conhecido como poço com completação inteligente (PCI), foi descrito por Goh et al. (2008) como aquele que possui tecnologia que permite o monitoramento remoto e o controle do fundo do poço, ambos pela superfície.

Os dispositivos instalados no tubo produtor, como os sensores e válvulas para controle da vazão, permitem o monitoramento das condições operacionais do poço (pressão, Ph da água etc.) e a projeção da distribuição dos atributos do reservatório (resistividade e impedância acústica). Vale ressaltar que também é possível controlar a entrada e saída dos fluxos através dos ajustes das válvulas em tempo real e de modo dinâmico (Glandt, 2003). Eles fornecem ao operador a habilidade de direcionar água e gás através do poço injetor quando necessário (Ranjith et al., 2017). A Figura 1 ilustra o funcionamento de um PCI com diferentes zonas de produção.

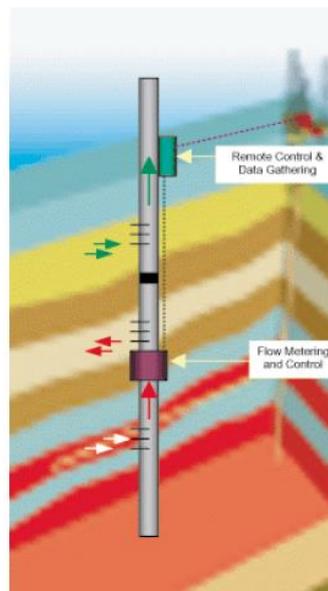


Figura 1: Representação do funcionamento do *Smart Well*

Fonte: Glandt, 2003

Os PCIs possuem a flexibilidade de controlar cada zona produtora de forma independente, o que permite uma rápida ação caso ocorra algum evento inesperado durante o desenvolvimento da produção de um campo de petróleo (Durlifky e Aziz, 2002).

Toda a tecnologia empregada nos PCI - incluindo técnicas, softwares e equipamentos para a leitura de novos dados e informações sísmicas – fornecem, na operação de produção de petróleo e gás, um maior controle de forma mais otimizada e precisa. Além disso, também é possível identificar novas áreas com potencial de exploração (Glandt, 2003).

Existem várias vantagens na implementação dos PCIs e a principal delas é a otimização da produção de petróleo e gás, que pode ser feita através da otimização da produção de cada poço, aumentando desta forma a produção do reservatório, ou através da redução dos custos operacionais no campo de petróleo, por exemplo, a redução dos custos com intervenções. Além disso, Ranjith et al. (2017) cita outras principais vantagens do PCI, como:

- Monitoramento da pressão e temperatura do fundo do poço em tempo real;
- Controle da produção pela superfície em cada zona para gestão do reservatório;
- Redução da produção não desejada de água e gás;
- Aumento da recuperação final e prolongamento da vida útil do poço;
- Teste de produção de cada zona sem intervenções e com o mínimo de paradas da produção.

Desta forma, é possível dizer que o poço inteligente permite o conhecimento de informações relevantes, o que proporciona o gerenciamento da produção de forma mais otimizada e precisa, dado o contínuo recebimento de novas informações a cada instante.

O objetivo principal da otimização é desenvolver uma estratégia de controle, ou seja, de ajustes dos dispositivos existentes no poço, de forma a maximizar o valor do campo, seja por meio do aumento da produção do reservatório ou pela redução de custos de operação do campo.

Embora o sistema de completção inteligente possua um custo superior ao de um poço convencional, se utilizado corretamente ele gera um benefício mais que

compensatório, principalmente com uso das informações futuras que reduzem incertezas (Ranjith et al., 2017). O aumento no investimento voltado para o uso de poços inteligentes precisa, portanto, ser superado pelo aumento do Valor Presente Líquido (VPL) a fim de justificar a sua aplicação.

Neste contexto, faz sentido utilizar os métodos de otimização *data driven* (em português: “orientada por dados”), através dos quais uma grande massa de dados (como, por exemplo, a produção em tempo real dos poços, temperatura, pressão, mapas geológicos etc.) é usada para realizar melhores previsões e otimização da produção no tempo, além de auxiliar a solucionar problemas de alta complexidade.

## 2.2

### Data driven

A abordagem *data driven* tem ganhado destaque devido a sua facilidade de implementação e por fornecer insumos para a tomada de decisão. Ele é o resultado da necessidade de criação de um sistema que foge dos modelos tradicionais e que proporciona uma análise rápida e robusta com a capacidade de processar o grande volume de dados e informações disponíveis a cada dia.

Wong e Wang (2003) descrevem que a abordagem *data driven* é capaz de descobrir vários padrões da base de dados sem depender de conhecimento prévio para fornecer evidências, bem como é capaz de lidar com vários cenários e objetivos flexíveis. Além disso, ela permite a extração de informações e conhecimento dos dados interativos com alta velocidade. Balaji et al. (2018) descrevem que métodos *data driven* são úteis para prever tendências futuras, prever comportamentos e responder perguntas que não são solucionadas pelos métodos tradicionais.

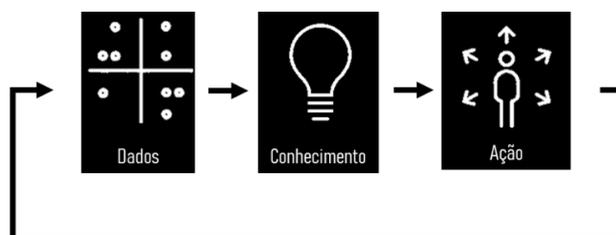


Figura 2: Representação *data driven*

Fonte: Figura elaborada pela autora

Segundo Stone (2007), a abordagem *data driven* se destaca dos modelos tradicionais (fundamentados em equações físicas e científicas) por ser menos complexa ao não exigir conhecimentos científicos e de engenharia muito específicos e por ser mais flexível, visto que não segue um modelo rígido de medidas e processos. Por serem baseados nos dados, os modelos *data driven* dependem dos conhecimentos das correlações e reconhecimento de padrões para fazer previsões, ou seja, são modelos que usam uma abordagem estatística com base na série histórica de dados (Temizel et al., 2017). Stone (2007) ressalta que esta também é uma fraqueza do modelo, pois, apesar do rápido aprendizado à medida que novos dados são disponibilizados, ele só irá funcionar para aqueles padrões aprendidos anteriormente por se tratar de uma técnica de reconhecimento de padrões.

No setor de petróleo e gás, a abordagem *data driven* em conjunto com o uso de algoritmos de *machine learning* e técnicas de *data mining* mostrou ter um importante papel na gestão dos reservatórios exploratórios e otimização da produção. Sabe-se que o processo de produção de óleo e gás é, por natureza, bastante complexo e envolve um grande volume de dados de caráter geológicos, geofísicos, petrofísicos e de engenharia de petróleo. Desta forma, fazer uma previsão e otimizar a produção em um reservatório de petróleo envolve um grande volume de dados e é o ponto chave no processo da tomada de decisão para se obter vantagens competitivas de mercado. Crnkovic-Friis e Erlandson (2015), por exemplo, usaram a abordagem *data driven* com algoritmos de *deep learning* para fazer uma modelagem geológica.

Stone (2007) descreveu exemplos de como a abordagem *data driven* para previsão foi vantajosa na British Petroleum quando usada em conjunto com as abordagens tradicionais. O autor ressalta também a importância da série histórica dos dados das operações para o sucesso das previsões. Temizel et al. (2017) discorre que a falta de uma base de dados histórica organizada de um poço limita a obtenção de melhores resultados.

Vale ressaltar que, além da obtenção de dados históricos, o primeiro passo para se construir um modelo *data driven* é a aplicação de técnicas de *data mining* com a seleção de atributos e intervalos relevantes. O *data mining* tem um importante papel no sucesso do modelo *data driven* e pode consumir bastante tempo para encontrar a melhor maneira de tratar e normalizar os dados, além de identificar as

correlações entre eles. Temizel et al. (2017), por exemplo, testou a significância de cada atributo através de uma regressão multivariada. Somente foram selecionados aqueles que tinham grande influência no resultado esperado, que, no caso, foi o desempenho do poço. Os autores ressaltam ainda que as técnicas de *data mining* foram bastante valiosas quando os mecanismos de monitoramento dos reservatórios falharam na decisão de uma estratégia ótima para desenvolvimento do campo devido ao grande número de incertezas.

De uma forma geral, a metodologia *data driven* tem tido crescente destaque no setor de óleo e gás, com grande foco na otimização das operações de produção *offshore*. A principal diferença é a combinação dos dados usados, o método *machine learning* e o número de poços monitorados no reservatório.

Dentre as principais técnicas de *machine learning* usadas no *data driven* estão as Redes Neurais Artificiais (RNA), os modelos de resistência a capacitância (MRC) e as metodologias híbridas que envolvem a combinação de métodos inteligentes como algoritmos genéticos, lógica *fuzzy*, redes neurais, dentre outras. Vale ressaltar que, devido a característica específica de cada poço (geologia, profundidade, pressão e etc), não existe uma metodologia única e seria improvável encontrar uma solução de fácil generalização.

Goh et al. (2007) usou a técnica *data driven* em duas situações. A primeira delas foi para estimar a previsão em tempo real da produção de petróleo, água e gás de cada poço de um reservatório, dada uma mudança na produção em um dos poços. Foi realizada uma otimização na produção dos demais poços para compensar a restrição de produção em um dos poços do reservatório.

A segunda técnica foi usada para estimar a produção por poço nos casos em que as informações sobre os testes de produção não foram fornecidas. Para este caso específico, o autor explica que a produção de um poço é dada através do seu teste de produção. Este teste é realizado periodicamente (a frequência pode variar de acordo com a demanda) e a produção é dada como constante de acordo com o resultado do teste até a próxima medição. Neste contexto, os modelos *data driven* são usados para estimar a produção por poço nos casos em que as informações dos testes individuais não são disponíveis, ou seja, só existem as informações de produção de todos os poços em conjunto.

Goh et al. (2008) descreve que os modelos *data driven* desenvolvidos a partir de dados de produção de poços de petróleo localizados no reservatório de

Champion West, fornecem, em tempo real, os dados de produção de petróleo, água e gás de cada poço, visando maior controle e melhor gestão do reservatório. Por meio do software chamado FieldWare Production Universe, o autor ressalta que o uso *data driven* para controle da produção do poço possui várias vantagens como, por exemplo, a previsão de produção (mesmo no caso de falha em algum sistema) e a simplicidade na sua abordagem. Em continuação a este artigo, Cramer e Goh (2009) descrevem vários estudos de caso e concluem que os benefícios desta abordagem superam os custos.

A previsão de produção de petróleo também foi feita por técnicas *data driven* através do processo de injeção de gás assim como descrito por Shi Jing et al. (2015) usando o algoritmo chamado de *Subspace Predictive Controller* (SPC).

Outra abordagem com dados de parâmetros geológicos, propriedades petrofísicas de formação e propriedades operacionais foi feita por Holdaway e Laing (2015). Assim como descreve os autores, o processo de projetar e implementar a completação de um poço é complexo e repleto de incertezas devido às condições físicas da subsuperfície. É neste contexto que a abordagem *data driven* através de redes neurais foi utilizada para criar um modelo descritivo da subsuperfície, aplicado na fase de design e implementação da completação de um poço para maximizar a produção de petróleo. Os autores concluem que os *insights* fornecidos pela abordagem *data driven* podem ser eficientes, intuitivos e fáceis de se obter, além de importantes para o processo de tomada de decisão.

Também usando métodos *data driven* e Redes Neurais Artificiais, Cao et al. (2016) descreve dois diferentes cenários para a previsão de produção. O primeiro é a produção de petróleo em um poço já existente com base em seu histórico. O segundo tem intuito de prever a produção de um novo poço com base na produção de poços próximos com uma geologia similar. Os autores concluem que, apesar do processamento de um grande volume de dados, a abordagem *data driven* é bastante promissora.

Em complemento a Redes Neurais Artificiais, Temizel et al. (2017) utilizou a técnica *data driven* usando *Capacitance-Resistance Models* (CRM). Durante a produção de petróleo é necessário manter a pressão no reservatório e, desta forma, se torna crucial a injeção de água à medida que o petróleo é bombeado. Sendo assim, a abordagem *data driven* foi usada neste artigo para realizar a otimização da

injeção e produção de água do reservatório de petróleo com o objetivo de maximizar a produção.

## Redes Neurais Artificiais e *deep learning*

As Redes Neurais Artificiais (RNAs) surgiram com o intuito de realizar o processamento da informação em uma estrutura que reproduza a estrutura de neurônios e as operações do cérebro humano através de um sistema computacional. As RNAs são capazes de realizar operações de aprendizado, associação, generalização e abstração (Haykin, 2007). Estes sistemas aprendem, de acordo com os exemplos dados, a realizar certas tarefas sem a necessidade de programação para uma função específica. As RNAs são bastante efetivas para lidar com dados que possuem relações não-lineares e apresentam um excelente desempenho em funções complexas nos mais diversos problemas do setor de óleo e gás (Balaji et al., 2018).

As RNAs são comumente usadas para realizar previsões, clusterizações e modelos de classificação. De uma forma geral, as RNAs aprendem a reconhecer padrões e características através do conjunto de dados para aprendizagem fornecido. Quando usadas para fins de previsão, este método de *machine learning* possui a principal vantagem de ser capaz de realizar uma estimativa mesmo quando as relações entre as variáveis não são bem definidas e são de difícil modelagem matemática.

As RNAs são compostas por camadas de neurônios interconectados e possuem a seguinte estrutura geral: (i) a camada de *input* (que são os dados de entrada); (ii) as camadas escondidas (com neurônios independentes); e (iii) a camada de saída (que fornece o resultado esperado). Cada camada efetua operações em seus neurônios e repassa os resultados para as camadas consecutivas.

As RNAs foram usadas, por exemplo, por Lei Sun e Lu (2010) e Boomer (1995) para fazer a previsão da produção de petróleo. Elas também foram usadas por Al-Fattah e Startzman (2001) para fazer previsões acerca da produção de gás.

### 3.1

#### **Redes *Multilayer Perceptron***

Existem diversos tipos de arquiteturas de redes neurais. A estrutura *Multilayer Perceptron* (MLP) talvez seja a mais amplamente conhecida e utilizada.

A estrutura da MLP é composta por uma camada de entrada, camadas escondidas de neurônios artificiais e a camada de saída. O elemento processador da MLP é o neurônio artificial. A camada de entrada possui o número de nós igual ao número de entradas da rede. As informações de entrada ou *inputs* ( $x$ ), chegam ao mesmo tempo aos neurônios, ou seja, realizam o processamento paralelo da informação (Haykin, 2007).

Os neurônios fazem conexões ou sinapses e a intensidade destas conexões é dada pelos pesos sinápticos ( $w$ ). Os *inputs* apresentados aos neurônios são multiplicados pelos pesos sinápticos. O neurônio, por sua vez, realiza o somatório destas multiplicações, resultando na sua ativação ou não. O processo de ativação ou não de cada neurônio é dado por sua função de ativação. Esta é essencial uma vez que introduz o componente de não linearidade às RNAs. Isso faz com que as RNAs possam aprender mais do que relações lineares entre as variáveis dependentes e independentes, sendo as funções de ativação mais comuns às funções sigmoid, gaussiana e tangente hiperbólica.

Pode ocorrer, também, a inclusão do bias de entrada ( $b$ ) - variável somada à função de ativação - visando aumentar o grau de liberdade desta função e permitindo que o neurônio apresente uma saída não nula (independente de se a entrada for nula). O neurônio, demonstrado na Figura 3, recebe diversos sinais de entrada e retorna apenas um sinal de saída, que poderá servir de entrada para os neurônios na camada posterior.

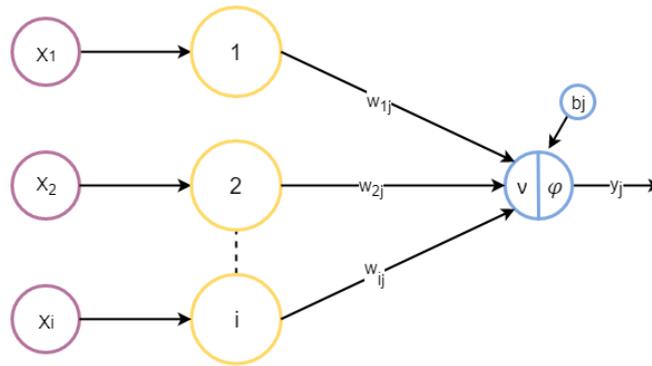


Figura 3: Representação do neurônio artificial

Fonte: Figura elaborada pela autora

Os neurônios são combinados em uma ou mais camadas escondidas - interconectadas através das sinapses - formando, portanto, a Rede Neural Artificial. A camada de saída contém, necessariamente, o número de neurônios igual ao número de *outputs* esperados. A RNA ajusta os seus pesos sinápticos de forma que eles representem a relação entre os *inputs* e os *outputs*, formando a memória associativa (Haykin, 2007). A Figura 4 mostra uma RNA simples com apenas uma camada escondida. Nela, pode-se notar a interconexão entre os neurônios e as camadas de entrada e saída. A maior parte das RNAs possuem o fluxo de informação unidirecional, sem a existência de ciclos, tendo uma arquitetura conhecida como *feedforward*.

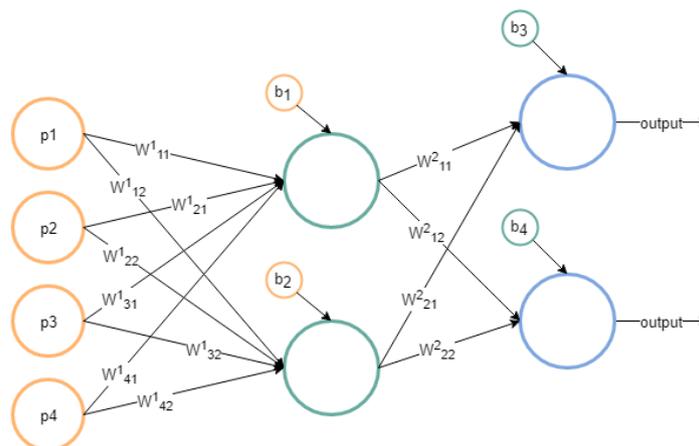


Figura 4: Representação da Rede Neural *Multilayer Perceptron (Feedforward)*

Fonte: Figura elaborada pela autora

As RNAs possuem a capacidade de aprendizagem e generalização a partir de experiências prévias, ou seja, a partir do treinamento obtido (Haykin, 2007). O treinamento da RNA pode ser supervisionado ou não supervisionado. A principal diferença é que, no supervisionado, a rede é treinada com um conjunto de entrada e de saída desejadas. Desta forma, toda vez que um conjunto de entrada é apresentado, a rede gera um conjunto de saída, comparando-o ao conjunto de saída desejado. Os pesos sinápticos e bias são ajustados e o processo é repetido até se alcançar um nível de acerto satisfatório.

Já no treinamento não supervisionado, a rede classifica os dados de entrada de forma autônoma, através do reconhecimento de padrões ou características similares. Isso é dado através do processo de competição e cooperação entre os neurônios.

Neste trabalho será utilizado somente o treinamento supervisionado, uma vez que o objetivo é treinar a rede para que ela seja capaz de realizar a previsão de produção de óleo, gás e água, conforme o exemplo apresentado.

## 3.2

### **Redes *deep learning***

As redes de aprendizado profundo ou *deep learning* consistem em uma técnica de aprendizado de máquina que consegue processar um grande volume de dados através de uma complexa arquitetura. As redes *deep learning* conseguem, com inúmeras camadas de rede (multicamadas), um melhor processamento das informações e melhor aprendizado. De uma forma geral, a rede *deep learning* aprende os padrões dos dados de entrada e cada camada extrai um tipo de informação específica (Goodfellow et. al., 2016).

A principal diferença ao se comparar redes *deep learning* com as redes neurais tradicionais (RNAs), é no tangente à arquitetura e ao processamento de dados. As RNAs possuem uma arquitetura mais simples, tendo dificuldade em processar um grande volume de dados e a análise de dados brutos como outras arquiteturas de *deep learning*.

Existem inúmeras arquiteturas das redes *deep learning* e a aplicação de cada uma delas depende unicamente do problema a ser tratado. Tendo em vista que o objetivo deste trabalho é realizar a previsão de séries temporais, foram utilizadas

arquitecturas do tipo Redes Neurais Recorrentes (RNR), mais precisamente a Rede Neural Recorrente Simples (RNNS), *long short-term memory* (LSTM) e *Gated Recurrent Unit* (GRU). Estas, devido ao componente de memória, têm a maior aplicabilidade para o problema em questão.

### 3.2.1

#### Redes Neurais Recorrentes (*Recurrent Neural Networks*)

As Redes Neurais Recorrentes (RNRs) são uma classe das redes *deep learning* na qual se usa o estado anterior para lidar com dados sequenciais. Como visto anteriormente, a rede MLP recebe sinais apenas da camada de entrada e processa estes sinais até chegar na camada de saída. Nas RNRs, por sua vez, a camada oculta recebe sinais tanto da camada de entrada quanto da camada oculta na iteração ocorrida no período anterior. Assim, as RNRs, para aprender adequadamente os pesos da rede, fazem uso tanto dos dados de entrada atuais, como do estado das etapas anteriores, processo chamado de *loop* (Althelaya et al., 2018). Este comparativo fica mais claro na Figura 5, abaixo.

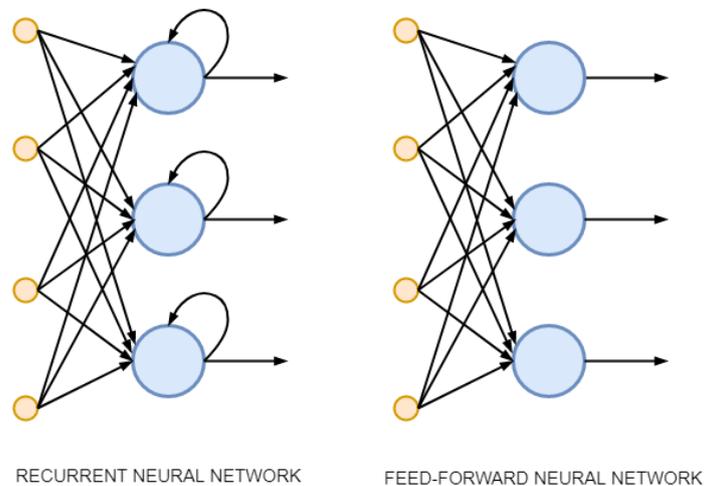


Figura 5 - Comparativo da estrutura da RNR e RNA *feedforward*

Fonte: Figura elaborada pela autora

O funcionamento da RNR pode ser descrito conforme a equação 1:

$$(1) h_t = f_W(h_{t-1}, x_t)$$

Onde  $h_t$  é o estado novo,  $f_W$  é a função com parâmetros,  $W$ ,  $h_{t-1}$  é o estado anterior e  $x_t$  o vetor de entrada no tempo  $t$ .

O funcionamento de uma RNR com três camadas ocultas (*hidden layers*) pode ser exemplificado de acordo com a Figura 6.

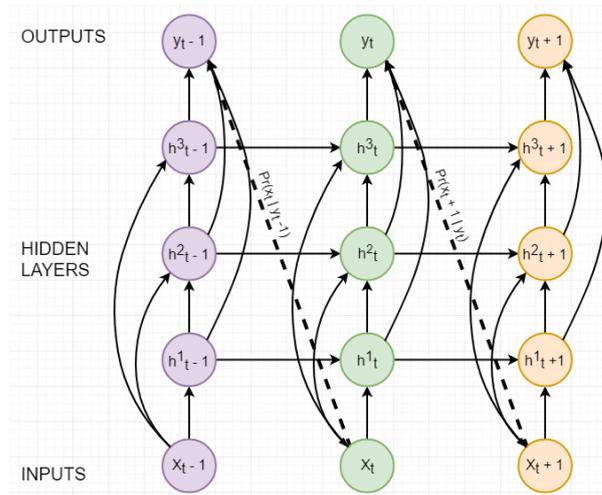


Figura 6- Arquitetura de uma RNR padrão com 3 camadas ocultas

Fonte: Figura elaborada pela autora

A rede RNR recebe a informação de entrada (*input*) em  $x_t$  e o resultado da probabilidade do processamento (*output*) no período anterior ( $t - 1$ ), ou seja,  $\Pr(x_t|y_{t-1})$ . Desta forma, cada interação utiliza o *output* da interação anterior, realizando uma forma do chamado *loop* ou corrente de redes neurais. Assim, quando as entradas passam pelas camadas ocultas  $h_t$ , é adicionada a informação do estado oculto do período anterior  $h_{t-1}W_h$ . Nota-se que as RNRs compartilham parâmetros através do tempo. A Figura 7 ilustra a estrutura interna de uma RNR padrão ou, em inglês, *Recurrent Neural Network (RNN)*, com o módulo repetitivo com uma função de ativação tangente hiperbólica ( $\tanh$ ).

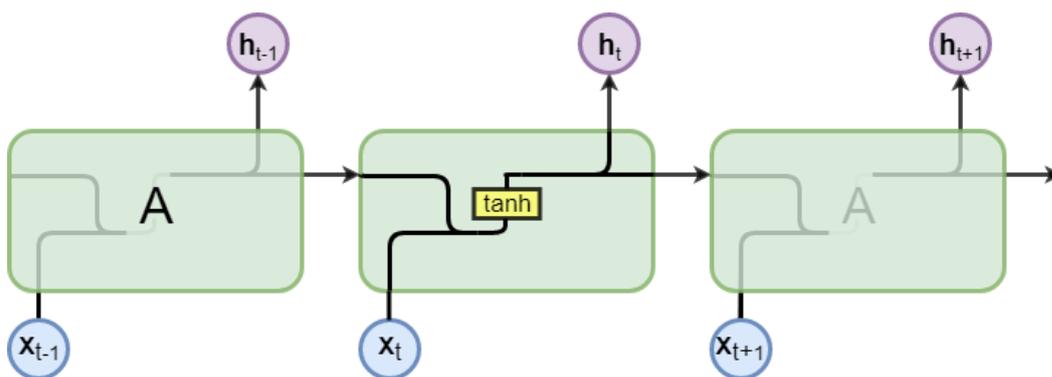


Figura 7 - Estrutura da unidade da RNR (RNN)

Fonte: Figura elaborada pela autora

A principal diferença da arquitetura das redes neurais convencionais MLP (*feedforward*) é o *loop*, uma vez que estas não memorizam toda a informação histórica ao realizar sua previsão. O *loop* faz com que as RNRs considerem como entrada não só o exemplo atual apresentado, como tudo o que lhe foi apresentado e processado anteriormente, já que ele é o componente de memória.

Esta capacidade de armazenar e aprender a partir de toda a informação apresentada sequencialmente permite às RNRs reconhecer um leque maior de padrões ou correlações subjacentes à sequência de dados e, com isso, executar previsões e tarefas que as redes MLP *feedforward* não conseguem. Essas correlações advindas da evolução do processamento são chamadas de dependências de longo prazo.

Para trabalhar com dados sequenciais, as RNRs dependem do movimento de *backpropagation* (ou retropropagação) do erro a cada espaço de tempo e do gradiente descendente. Assim como explica Zhang et al. (2018), o grande gargalo da RNR é que, para uma sequência muito grande, o gradiente pode tender ao infinito ou convergir para zero, uma vez que a evolução do erro retropropagado depende exponencialmente do tamanho dos pesos (Hochreiter e Schmidhuber, 1997). Para solucionar esta questão, foram desenvolvidas as redes LSTM e GRU, que, ao ampliar a memória para processar maiores sequências na entrada, conseguem prevenir que os erros retropropagados desapareçam ou tendam para o infinito (Hochreiter e Schmidhuber, 1997).

De uma forma geral, todas as RNRs possuem uma cadeia de módulos repetitivos de rede neural em uma rede simples padrão. Como visto na Figura 7, esta estrutura conta com apenas uma função tangente hiperbólica ( $\tanh$ ).

Nas redes LSTM e GRU, ao invés de um único módulo de repetição, existem portões em uma estrutura chamada de *Gated Recurrent Neural Network*. Estes portões (*gates*) são unidades especiais que mantêm o fluxo do erro constante, permitindo o ajuste de pesos. Os detalhes e especificidades das duas redes serão vistos mais adiante nos itens 3.2.2 e 3.2.3.

### 3.2.2

#### Redes *long short-term memory* (LSTM)

Existem várias arquiteturas das unidades das redes LSTM. A mais comum é ilustrada pela Figura 8.

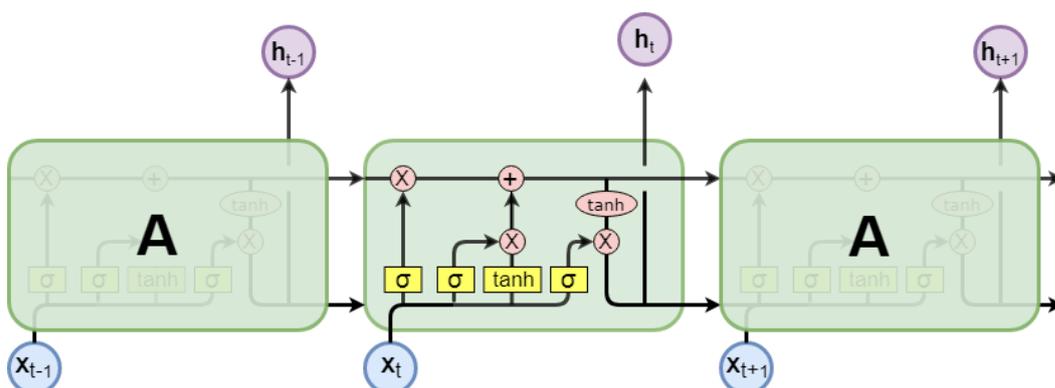


Figura 8 - Estrutura da unidade da rede LSTM

Fonte: Figura elaborada pela autora

A chave da LSTM é o estado celular ( $C_t$ ) que serve como uma esteira transportadora da informação pela qual percorre toda a cadeia com apenas algumas interações lineares. A informação pode fluir através dela sem sofrer alterações (Hochreiter e Schmidhuber, 1997).

A LSTM possui quatro camadas de interação compostas por um estado celular (a parte de memória da unidade) e três portões: uma porta de entrada (*Input*), uma porta de saída (*Output*) e uma porta de esquecimento (*Forget*).

O portão de esquecimento ou *Forget* ( $f$ ) é uma camada sigmoide na qual é estabelecido qual valor deve ser “esquecido”. O funcionamento deste portão é dado pela equação 2:

$$(2) f_t = \sigma(w_f \cdot [h_{t-1}, x_t] + b_f)$$

Onde:  $f_t$  representa o portão de esquecimento,  $w_f$  os pesos,  $h_{t-1}$  a saída do período anterior,  $x_t$  a entrada do período atual e  $b_f$  o *bias*.

O portão de esquecimento é responsável por olhar cada novo input e decidir quais informações de períodos anteriores devem ser removidas (ou esquecidas). A saída é uma variável entre 0 e 1, onde: “1” significa que valor deve ser mantido completamente e “0” indica que valor deve ser esquecido completamente. A Figura 9 ilustra o funcionamento deste portão.

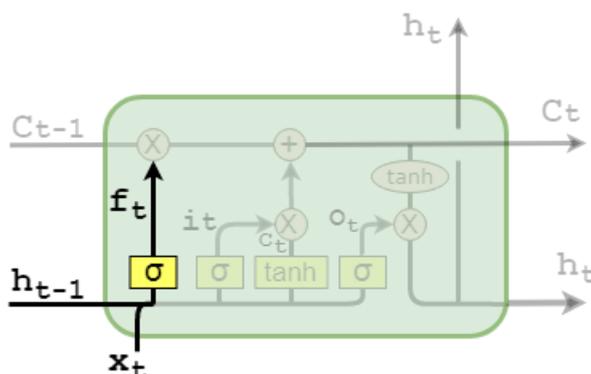


Figura 9- Portão de esquecimento (*Forget*)

Fonte: Figura elaborada pela autora

O portão de entrada ou *Input* ( $i$ ) é a camada sigmoide que decide quais valores do período atual são importantes de serem lembrados. O funcionamento deste portão é dado pela equação 3:

$$(3) i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i)$$

Onde:  $i_t$  representa o portão de entrada,  $w_i$  os pesos,  $h_{t-1}$  a saída do período anterior,  $x_t$  a entrada do período atual e  $b_i$  o *bias*.

Os valores passam por uma função tangente hiperbólica ( $\tanh$ ) que cria um vetor de candidatos (Candidate) ( $\tilde{c}_t$ ) a serem adicionados na memória de longo

prazo e fazer a atualização do estado celular (Hochreiter e Schmidhuber, 1997). Este processo é ilustrado na Figura 10 e seu funcionamento ocorre conforme a equação 4:

$$(4) \tilde{c}_t = \tanh(w_c \cdot [h_{t-1}, x_t] + b_c)$$

Onde:  $\tilde{c}_t$  representa o vetor de candidatos,  $w_c$  os pesos,  $h_{t-1}$  a saída do período anterior,  $x_t$  a entrada do período atual e  $b_c$  o *bias*.

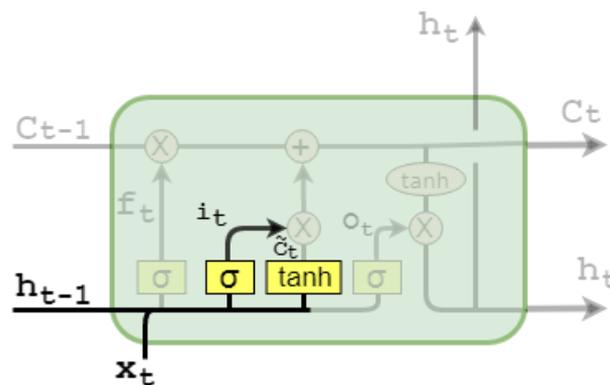


Figura 10 - Portão de entrada (*Input*) e de candidatura (*Candidate*)

Fonte: Figura elaborada pela autora

A atualização do estado celular de  $C_{t-1}$  para  $C_t$  acontece conforme a equação 5, através da soma de duas multiplicações. A primeira parte é a multiplicação da saída do portão de esquecimento  $f_t$  pelo estado celular antigo ( $C_{t-1}$ ). A segunda é a multiplicação dos novos candidatos  $i_t$  pela escala na qual o estado celular deve ser atualizado  $\tilde{c}_t$ .

$$(5) C_t = f_t * C_{t-1} + i_t * \tilde{c}_t$$

A ilustração da atualização do estado celular pode ser vista na Figura 11.

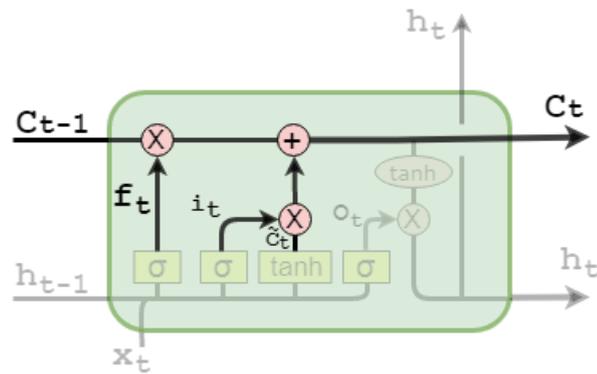


Figura 11- Atualização do estado celular

Fonte: Figura elaborada pela autora

Por fim, o portão de saída ou *Output* ( $o$ ) é o portão no qual se determina qual parte da entrada ( $o_t$ ), através de uma função sigmoid, deve ir para a saída ( $h_t$ ), com base no estado celular atualizado ( $C_t$ ) e em uma função tangente hiperbólica ( $\tanh$ ). O funcionamento deste portão é dado pela equação 6.

$$(6) o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o)$$

Ressaltando que o estado celular passa por uma função tangente hiperbólica ( $\tanh$ ) para que os valores fiquem no intervalo de -1 à 1 para serem multiplicados pela entrada  $o_t$ , a saída  $h_t$  irá conter apenas as partes que realmente interessam no aprendizado da rede, conforme a equação 7. A ilustração do portão de saída pode ser visto na Figura 12.

$$(7) h_t = o_t * \tanh(c_t)$$

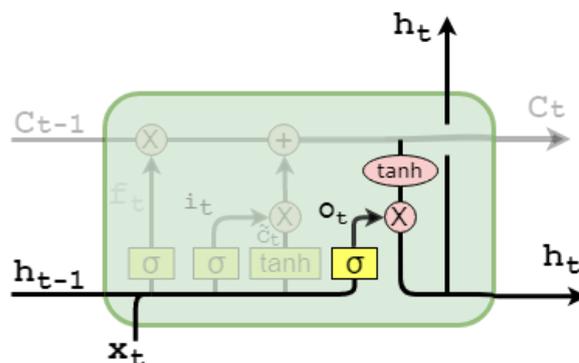


Figura 12 - Portão de saída (*Output*)

Fonte: Figura elaborada pela autora

É importante destacar que cada portão tem o funcionamento de uma rede neural com sua respectiva função de ativação e matriz de pesos.

A previsão de produção já foi testada por Sun et al. (2018) usando LSTM e sua eficácia foi comparada pelos autores, frente a outros métodos como o *Decline Curve Analysis* (DCA), devido à significativa redução do erro. Além disso, o *deep learning* foi considerado um método de modelagem para *data driven* bastante efetivo (Zhang et al., 2018).

### 3.2.3

#### Redes *Gated Recurrent Unit (GRU)*

A rede GRU foi proposta por Cho et. al. (2014) e pode ser considerada como uma versão atualizada e mais compacta da rede LSTM na medida em que demanda menos parâmetros e um número menor de portões (*gates*). De uma forma geral, a GRU não possui o estado celular e usa o estado oculto para transferir a informação. A GRU também não possui o portão de saída (*Output*) e não possui a estrutura do estado celular como a estrutura da LSTM (Cho et. al, 2014).

A estrutura da rede GRU pode ser observada na Figura 13.

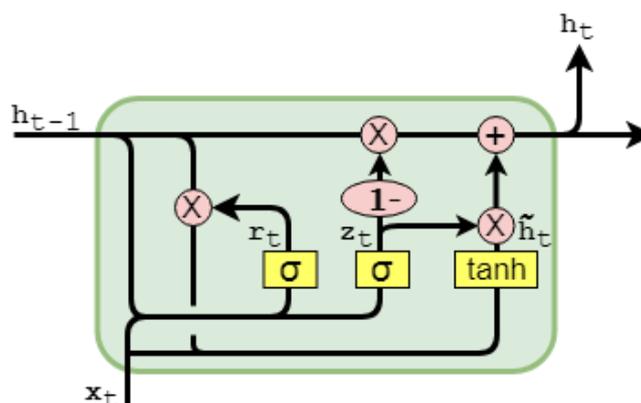


Figura 13- Estrutura da unidade da rede GRU

Fonte: Figura elaborada pela autora

A GRU é composta por dois portões: o de *update* e o de *reset*.

O portão de *update* - ou atualização ( $z_t$ ) - é responsável por definir quanta informação passada é preciso manter na memória nos períodos seguintes. Este portão é uma versão combinada dos portões de entrada e esquecimento da LSTM. Estes portões funcionam através da multiplicação das entradas ( $x_t$ ) e ( $h_{t-1}$ ) pelos

seus respectivos pesos ( $W$ ). Estes passam por uma função sigmoidm, resultando em um valor entre “0” e “1”. Seu funcionamento é dado pela equação 8:

$$(8) z_t = \sigma (w_z \cdot [h_{t-1}, x_t])$$

Onde:  $z_t$  representa o portão de atualização,  $w_z$  os pesos,  $h_{t-1}$  a saída do período anterior e  $x_t$  a entrada do período atual.

O portão de *reset* ( $r_t$ ) é responsável por definir o quanto da informação do período passado deve ser esquecida. A fórmula é parecida com a usada no portão de atualização, porém, ela usa pesos diferentes uma vez que tem o objetivo de esquecer ao invés de manter a informação. A função também passa por uma função sigmoid, conforme a equação 9:

$$(9) r_t = \sigma (w_r \cdot [h_{t-1}, x_t])$$

Onde:  $r_t$  representa o portão de reconfiguração,  $w_r$  os pesos,  $h_{t-1}$  a saída do período anterior e  $x_t$  a entrada do período atual.

A atualização da memória celular ( $\tilde{h}_t$ ) passa por uma função tangente hiperbólica ( $\tanh$ ) e ocasiona na combinação da nova entrada com a memória do período passado, feita através da informação que passou pelo portão *reset*, mantendo, assim, somente a informação do passado que é relevante. Isso pode ser visto na equação 10.

$$(10) \quad \tilde{h}_t = \tanh (w \cdot [r_t * h_{t-1}, x_t])$$

Por fim, a saída do período atual ( $h_t$ ), demonstrada na equação 11, depende da atualização da memória celular dada por ( $\tilde{h}_t$ ) e da informação mantida dos períodos anteriores.

$$(11) \quad h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

Ambas as redes LSTM e GRU são consideradas estado da arte no processamento de linguagem natural, reconhecimento de fala etc. O desempenho delas depende das características da aplicação. Por ser mais simples, a rede GRU é muitas vezes mais rápida para treinar e precisa de pesos para atualizar.

Neste trabalho, ambas as redes, LSTM e GRU, apresentam o benefício de memorizar longas séries históricas da posição das válvulas e produção

correspondente. Com isso, elas fornecem uma melhor estimativa para o período seguinte da produção de óleo, gás e água.

### 3.2.4

#### Arquiteturas bidirecionais

A arquitetura de redes bidirecionais foi proposta inicialmente por Schuster e Paliwal (1997) e usa, no treinamento, toda a informação disponível no passado e futuro de janela de tempo. Com a utilização da informação nas duas direções (passado e futuro), é possível otimizar a função objetivo sem a necessidade de atrasos (*delays*) para incluir a informação futura como a RNR unidirecional padrão (Schuster e Paliwal, 1997).

A camada bidirecional consiste em uma arquitetura composta por duas camadas conectadas. Cada uma delas realiza operações com uma sequência de entrada em determinada direção. Ou seja, uma primeira camada executa as operações seguindo a mesma direção de fluxo da sequência de dados enquanto a segunda camada aplica as operações da mesma sequência no sentido reverso (Althelaya et al., 2018). O exemplo da arquitetura bidirecional na rede LSTM pode ser visto na Figura 14. Note que as saídas das duas camadas não conectam entre si. Para isso, existem diferente metodologias visando combinar estes resultados no aprendizado da rede.

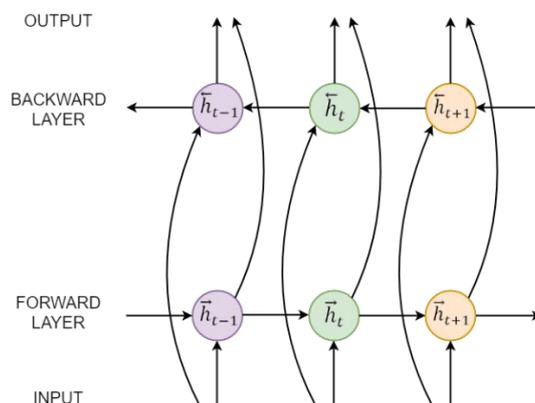


Figura 14 – Arquitetura da LSTM Bidirecional

Fonte: Figura elaborada pela autora

Duas camadas de LSTM podem ser empilhadas usando duas direções para o melhor aprendizado da sequência através de uma arquitetura bidirecional (LSTMB). O mesmo se aplica à rede GRU (GRUB) e Rede Neural Recorrente Simples (RNNSB).

Como proposto por Zhang et al. (2018), para fazer a previsão de produtividade de produtos para unidades de craqueamento catalítico em leito fluidizado, foi testado neste trabalho o uso da arquitetura bidirecional na previsão. De acordo com os autores, uma camada bidirecional na rede LSTM (ou no caso deste trabalho GRU ou RNNS), permite lidar com a dependência de longo prazo ao integrar uma estrutura profunda bidirecional com a estrutura da Rede Neural Recorrente (LSTM, GRU ou RNNS).

A aplicação da arquitetura bidirecional foi testada em todas as redes *deep learning* (do tipo Redes Neurais Recorrentes) e os resultados foram comparados com arquiteturas de fluxo unidirecional.

## **Smart proxy e deep smart proxy**

A previsão de produção de petróleo dos reservatórios é uma das tarefas mais difíceis e importantes na engenharia de petróleo, uma vez que, segundo Kohler (2013), a previsão permite, além de otimizar a produção, quantificar as reservas de forma que se possa realizar a avaliação e priorização de projetos de exploração ou desenvolvimento. Os simuladores computacionais de reservatórios são utilizados há anos nesta tarefa. No entanto, eles requerem um grande poder de processamento para obter resultados em tempo factível, além do alto custo envolvido. Zangl et al. (2006) ressalta que, para grandes reservatórios com inúmeras variáveis incertas, o esforço computacional para estimar todas as possíveis soluções se torna praticamente impossível, além do tempo inviável para estimar todas as combinações possíveis. Desta forma, os aproximadores ou modelos *proxy*, surgem como uma alternativa para diminuir o tempo e o custo necessário para a realização das simulações, além de otimizar o esforço computacional de forma a não exigir um grande número de simulações para que este não seja um fator limitante no processo de otimização.

Os modelos *proxy* são amplamente usados no setor de petróleo. Zangl et al. (2006), por exemplo, usou Redes Neurais Artificiais para construir um modelo *proxy* visando a otimização de produção com algoritmos genéticos. As Redes Neurais Artificiais também foram usadas por Kohler (2013) para o construir o modelo *proxy*. De forma geral, os modelos *proxy* são uma alternativa barata quando a simulação completa e detalhada dos reservatórios não se mostra viável (He et al., 2016). Os modelos *proxy* devem, portanto, captar as tendências não lineares de produção através de uma amostra da simulação (Zangl et al., 2006) e devem replicar o comportamento do reservatório com uma acurácia alta e com baixo esforço e tempo computacional (He et al., 2016).

Uma fraqueza desses modelos frente aos simuladores é que eles são menos precisos e, portanto, tem uma maior margem de erro (Zangl et al., 2006). Além disso, ficam restritos ao aprendizado da amostragem, ou seja, não realizam boas previsões quando o valor extrapola o intervalo da amostra.

Com a evolução das técnicas de *machine learning*, os modelos *proxy* vêm sendo capazes de realizar previsões com menores erros e com maior capacidade de generalização. Diante disso, surgiram os modelos *smart proxy* que, como explica He et al. (2016), são diferentes de um modelo *proxy* convencional, pois utilizam, de forma combinada, a metodologia *data driven* com técnicas de inteligência artificial e *machine learning*. Neste contexto, as Redes Neurais Artificiais aprendem os padrões das variáveis e imitam o comportamento do reservatório com um processamento em alta velocidade e alta acurácia (Mohaghegh et al., 2015). A capacidade de aprendizado seria, portanto, a principal característica que difere o modelo *smart proxy* do modelo *proxy*, uma vez que este último seria baseado apenas em métodos estatísticos e matemáticos que visam replicar o simulador de reservatório com menor tempo e esforço computacional, porém com um erro maior.

O *smart proxy* tem diversas aplicações na engenharia de petróleo e foi usado por He et al. (2016), por exemplo, para realizar uma correspondência dos valores históricos da produção mensal de petróleo e água com o uso de redes neurais. Foi também usado por Mohaghegh et al. (2015) para otimizar a produção de petróleo em um grande reservatório de petróleo nos Emirados Árabes Unidos e por Alenezi e Mohaghegh (2017) para replicar as saídas do simulador de reservatório usando técnicas de data mining e Redes Neurais Artificiais.

Neste trabalho, além do modelo *smart proxy* com o uso de redes neurais (do tipo Multilayer Perceptron), foi também proposto o modelo *deep smart proxy*, pelo qual são utilizadas redes *deep learning* do tipo Redes Neurais Recorrentes para aprendizado. Esta diferenciação se tornou necessária devido às diferenças na arquitetura das redes *deep learning* do tipo Redes Neurais Recorrentes em comparação as redes neurais convencionais Multilayer Perceptron. Ambos os modelos foram propostos neste trabalho como forma de estratégia no controle de produção de petróleo em um reservatório de petróleo com um poço produtor já existente. Os experimentos com os modelos *smart proxy* e *deep smart proxy* foram realizados com base na modelagem de dados de um reservatório sintético que possui três camadas com diferentes permeabilidades. As três válvulas estão localizadas no poço injetor, sendo uma para cada completção conforme Figura 15.

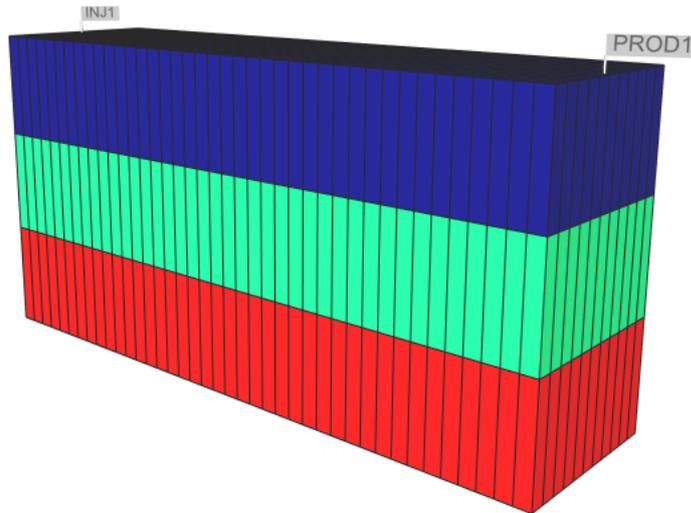


Figura 15: Modelo do reservatório sintético

Fonte: Figura elaborada pela autora

Portanto, para se ter uma previsão da curva de produção de óleo, água e gás, foram realizados diversos testes com o intuito de avaliar qual método de *machine learning* apresenta a menor raiz do erro médio quadrático (RMSE). Desta maneira, a estimativa pode auxiliar na tomada de decisão de forma que seja possível:

- 1) decidir no período atual a melhor configuração para a válvula no período seguinte, ou seja, dada a configuração das válvulas hoje ( $t$ ), qual a configuração que no próximo período ( $t+1$ ) irá maximizar a produção;
- 2) comparar a produção real com a estimativa que foi realizada para o período  $t$ . Esta comparação pode fornecer insumos sobre a situação da produção, uma vez que, caso esteja ocorrendo algum problema operacional é razoável esperar que isto irá se refletir numa diferença cada vez maior entre o produzido e o estimado;
- 3) analisar o impacto futuro na produção de óleo e gás ao perfurar outro poço produtor próximo ao mesmo reservatório;
- 4) auxiliar na análise do período ótimo de abandono quando o tempo de vida do reservatório estiver no fim, visto que a previsão de produção de óleo e gás tem a tendência de ser cada vez menor.

## 5

### **Previsão com dados simulados - modelos *smart proxy* e *deep smart proxy***

#### 5.1

##### **Base de dados**

Para gerar as estimativas de produção de óleo, gás e água foi utilizado o software de simulação de reservatório chamado *Computer Modelling Group* (CMG). Este software faz simulações de produção a partir das condições de estado geológico, número de poços e condições de pressão. Embora seja possível realizar simulações de vários estados geológicos para o mesmo poço, por simplificação, este estudo considerou apenas um.

As simulações são de apenas um poço produtor em que existem três válvulas que possuem configurações binárias independentes, podendo cada válvula estar aberta (1) ou fechada (0). Vale ressaltar que cada válvula se encontra em uma profundidade do poço e para simplificar de forma a reduzir o tempo computacional, a configuração destas válvulas pode ser alterada somente a cada seis meses. Desta forma, o conjunto das configurações destas válvulas resulta na estimativa por mês da produção média diária de óleo, gás e água. A configuração destas válvulas foi gerada aleatoriamente.

A base de dados é composta por 3119 simulações de diferentes configurações das válvulas e as produções resultantes (de óleo, gás e água) no período total de 10 anos. Desta forma, tem-se 3119 séries históricas, divididas entre o conjunto de treinamento, validação e teste.

Todos os testes foram realizados usando Python na versão 3.6 no *Cluster* do Intel AI DevCloud com as configurações da Tabela 1.

Tabela 1 – Configuração dos Nós do Cluster do IntelDevCloud

Intel AI DevCloud	Configuração dos nós do <i>Cluster</i>
	Características
Família	Intel® Xeon® Processor
Arquitetura	Skylake architecture (Intel Xeon Scalable Processors Family)
Modelo	Intel® Xeon® Gold 6128 CPU
Tamanho da memória	Ram memory: 96 GiB
Networking	Gigabit Ethernet Interconnect

Fonte: Tabela elaborada pela autora

A base de dados foi dividida da seguinte forma: 80% para treinamento e validação e 20% para teste. Por conter diferentes variáveis e magnitudes, a base de dados foi normalizada antes do treinamento, validação e teste.

## 5.2

### **Smart proxy – testes e resultados**

O objetivo principal dos testes os métodos de *machine learning* é analisar se a rede consegue aprender a estimar a produção de óleo, gás e água, com base em uma sequência temporal com o histórico da posição das três válvulas e produção mensal. Os testes com as RNAs com a rede *Multilayer Perceptron* (MLP), que neste trabalho é chamado de *smart proxy*, são considerados o estado da arte da literatura, e é uma metodologia que já foi amplamente aplicada para fazer previsão no setor de óleo e gás e está sendo testado neste trabalho para fins de comparação.

Foram testadas diferentes sequências temporais, ou seja, variou-se o número de meses que elas foram utilizadas como *input* para o aprendizado da rede. Convém, portanto, observar que as entradas são vetores e que foram testados modelos com 2 camadas escondidas. A arquitetura da rede MLP considerada foi a da Figura 16, que, por simplicidade, mostrou apenas uma camada escondida.

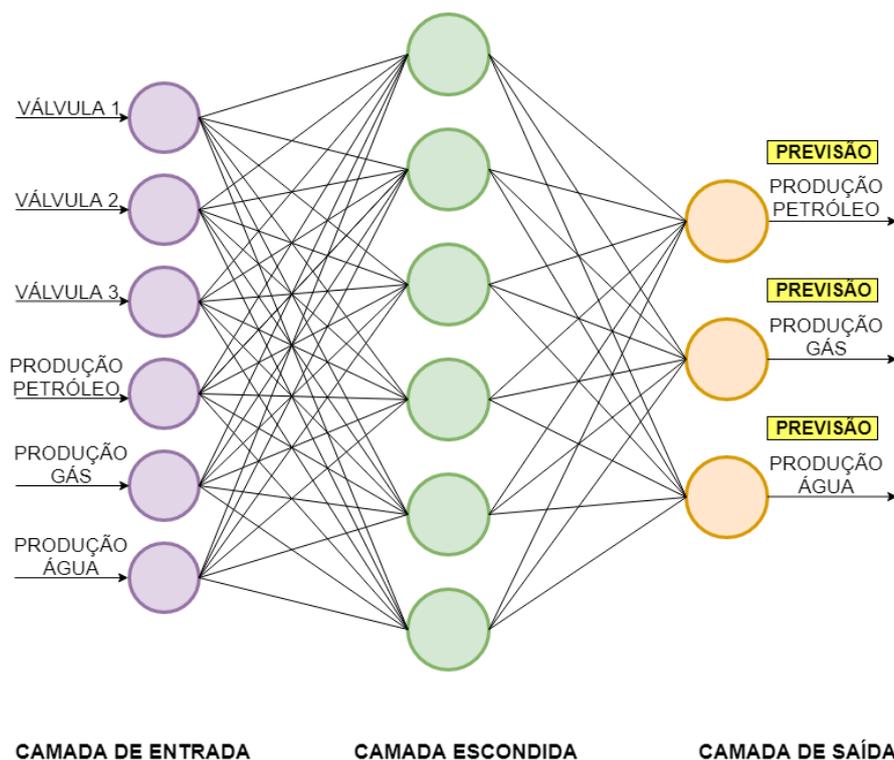


Figura 16: Arquitetura da rede MLP no modelo *smart proxy*

Fonte: Figura elaborada pela autora

Para evitar o problema de *overtraining* descrito por Weiss et al. (2002), foi aplicado o método de parada antecipada ou *early stopping*. Buscou-se uma configuração que forneça o melhor desempenho através do menor RMSE total. Os pesos foram inicializados de forma aleatória e os demais parâmetros básicos da MLP avaliados neste estudo são apresentados na Tabela 2.

Tabela 2 – Parâmetros das Redes Neurais Artificiais (MLP)

<b>Parâmetros das Redes Neurais</b>	
Tipo	MLP
Topologia	2 camadas escondidas
Neurônios (em cada camada escondida)	Variando de 32 a 246
Função de ativação	Tangente hiperbólica (tanh)
Função de saída	Tangente hiperbólica (tanh)
Nº máximo de épocas*	200
*Critério de parada definido pelo erro do conjunto de validação ou pelo número máximo de épocas.	

Fonte: Tabela elaborada pela autora

A Tabela 3 mostra os resultados dos testes realizados com a rede MLP em ordem de melhor desempenho. A métrica para determinar o melhor desempenho da rede foi através da menor raiz quadrada do erro médio quadrático (RMSE<sup>1</sup>). Ressaltando que também foi estimado a métrica de média absoluta do erro (MAE<sup>2</sup>) por considerar que se trata de uma importante medida na análise da magnitude e dimensão do erro. O RMSE foi escolhido como medida de erro, pois é uma métrica que penaliza erros maiores e evita o uso de valores absolutos. Como o propósito do trabalho é encontrar uma rede que forneça a melhor previsão para o petróleo, gás e água em conjunto, o melhor desempenho foi dado pelo RMSE total. Vale ressaltar que foram estimados os erros MAE e RMSE separados por tipo de produção. O valor percentual (ou normalizado) por produção foi estimado apenas para o RMSE<sup>3</sup>, uma vez que o MAE (que no caso seria o MAPE – média absoluta do erro percentual) não pôde ser calculado já que os valores estimados quando iguais a zero fazem com que o MAPE tenha valor infinito.

Tabela 3 – Resultados dos testes *smart proxy* usando Redes Neurais *Multilayer Perceptron*

Modelo	Épocas (Early stopping)	Batch Size	Neurônios	Sequência	RMSE Petróleo	RMSE Gás	RMSE Água	RMSE Total	%RMSE Petróleo	%RMSE Gás	%RMSE Água	%RMSE Total	MAE Petróleo	MAE Gás	MAE Água	MAE Total
MLP	173	64	64/128	10	27,83	0,77	4,51	16,28	2,39%	2,34%	0,86%	1,87%	9,99	0,29	1,20	3,83
MLP	73	64	64/128	18	28,17	0,77	5,02	16,53	2,43%	2,35%	1,09%	1,96%	9,57	0,26	2,09	3,97
MLP	97	64	128/246	10	28,94	0,79	4,77	16,94	2,49%	2,42%	0,91%	1,94%	8,74	0,24	2,18	3,72
MLP	160	64	32/64	12	28,87	0,79	5,18	16,94	2,48%	2,41%	1,08%	1,99%	9,80	0,26	1,80	3,96
MLP	169	64	30/60	18	29,33	0,80	4,79	17,17	2,53%	2,46%	1,04%	2,01%	9,59	0,27	1,30	3,72
MLP	48	64	128/246	12	30,16	0,83	4,50	17,61	2,59%	2,52%	0,94%	2,02%	10,67	0,29	1,36	4,10
MLP	104	64	64/128	8	30,14	0,83	5,99	17,75	2,59%	2,52%	1,15%	2,09%	10,17	0,27	3,18	4,54
MLP	75	64	64/128	14	30,31	0,83	5,09	17,75	2,61%	2,53%	1,06%	2,06%	11,63	0,31	2,02	4,65
MLP	58	32	64/128	12	30,85	0,84	5,26	18,07	2,65%	2,57%	1,09%	2,11%	11,36	0,30	2,01	4,56
MLP	75	128	32/64	12	31,51	0,87	4,90	18,45	2,71%	2,64%	1,02%	2,12%	10,18	0,28	1,55	4,00

Fonte: Tabela elaborada pela autora

Nota-se que a melhor configuração se encontra em destaque na primeira linha. Nela, foram testadas duas camadas escondidas com 64 e 128 neurônios ao

<sup>1</sup> RMSE é calculado como a raiz do erro médio quadrático da diferença entre a previsão e o valor real, ou seja,  $RMSE = \sqrt{\frac{1}{n} \sum_{j=1}^n (y_j - \hat{y}_j)^2}$ .

<sup>2</sup> MAE é calculado como a média ponderada absoluta do erro, ou seja,  $MAE = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j|$ .

<sup>3</sup> O RMSE normalizado é calculado dividindo o valor do RMSE pela subtração do maior e menor valor, ou seja,  $NRMSE = \frac{RMSE}{y_{max} - y_{min}}$ .

longo de 10 meses. Nesta configuração, observa-se que o RMSE foi de 16,28, o que representa 1,87% de erro. Ao comparar as configurações testadas, é possível perceber uma variação do erro RMSE de 12% e MAE de 4% entre os desempenhos.

### 5.3

#### **Deep smart proxy – Testes e Resultados**

Neste trabalho foram realizados testes com três tipos de algoritmos *deep learning* do tipo Redes Neurais Recorrentes, são eles: *long short-term memory* (LSTM), *Gated Recurrent Unit* (GRU) e uma Rede Neural Recorrente Simples (RNNS). Conforme explicitado anteriormente, os testes com a *deep smart proxy* foram realizados usando Python na versão 3.6 e no cluster do Intel AI Dev Cloud.

Visando evitar o problema de *overfitting*, assim como Zhang et al. (2018), foi utilizado a metodologia de parada antecipada ou *early stopping*. Ao treinar o modelo, o erro quadrático médio (MSE) foi utilizado como função para medir a perda, ou seja, uma medida para entender o quão distante estão os dados através da diferença entre o valor de produção real e o previsto. Os dados foram divididos em conjuntos de treinamento, validação e testes, sendo 80% para treinamento e validação e 20% para teste.

O método de otimização em todos os testes realizados neste trabalho foi o Adam, uma vez que este apresenta uma vantagem frente aos outros métodos estocásticos de otimização, assim como executado por Zhang et al. (2018). De acordo com Kingma e Ba (2015), o método combina as vantagens de dois métodos de otimização: a capacidade do AdaGrad de lidar com gradientes esparsos e a capacidade do RMSProp de lidar com objetivos não estacionários. Além disso, os autores afirmam que o Adam é mais simples para implementar e requer pouca memória, sendo, portanto, considerado o mais robusto quando a base de dados e o número de parâmetros são grandes.

Como forma de visualizar melhor os resultados, os testes com cada algoritmo foram separados em tabelas diferentes da seguinte forma: Tabela 4 possui os resultados dos testes com RNNS, Tabela 5 possui os resultados dos testes com LSTM e Tabela 6 possui os resultados dos testes com GRU. Foram realizados testes com camadas bidirecionais. Estes foram destacados com a letra “B” na coluna com as descrições dos métodos de *machine learning* (ex. GRUB, LSTMB e RNNSB).

Assim como nos testes com a rede MLP, os testes com algoritmos *deep learning* do tipo RNR estão em ordem de desempenho e, portanto, o melhor resultado de cada um deles se encontra em destaque na primeira linha. É preciso ressaltar que o melhor desempenho foi estimado com base no menor RMSE total.

Tabela 4 – Resultados dos testes *deep smart proxy* com a Rede Neural Recorrente Simples (RNNS) e bidirecional (RNNSB)

Modelo	Épocas (Early stopping)	Batch Size	Unidades	Sequência	RMSE Petróleo	RMSE Gás	RMSE Água	RMSE Total	%RMSE Petróleo	%RMSE Gás	%RMSE Água	%RMSE Total	MAE Petróleo	MAE Gás	MAE Água	MAE Total
RNNB	42	128	128/256	14	15,21	0,42	5,84	9,41	1,31%	1,28%	1,12%	1,23%	7,54	0,21	3,54	3,76
RNNB	29	64	64/128	14	16,33	0,45	6,43	10,13	1,40%	1,37%	1,23%	1,34%	5,68	0,16	2,10	2,65
RNNB	21	64	64/128	12	17,24	0,47	6,02	10,55	1,48%	1,43%	1,15%	1,36%	8,41	0,22	3,11	3,92
RNNB	20	128	128/256	18	21,09	0,57	9,70	13,41	1,81%	1,75%	1,86%	1,81%	11,10	0,30	4,64	5,34
RNNB	25	128	128/256	12	24,24	0,66	9,71	15,08	2,08%	2,00%	1,86%	1,98%	15,87	0,42	5,04	7,11
RNNB	8	256	256/512	18	32,20	0,83	15,00	20,51	2,77%	2,53%	2,87%	2,72%	20,62	0,52	9,53	10,22
RNN	15	256	256/512	18	50,71	1,41	15,07	30,55	4,36%	4,30%	2,85%	3,85%	31,02	0,87	7,80	13,23
RNN	19	256	256/512	14	51,17	1,35	21,10	31,97	4,40%	4,12%	4,04%	4,18%	33,16	0,81	16,90	16,95
RNNB	8	256	256/512	14	63,41	1,77	21,22	38,62	5,45%	4,41%	4,06%	4,97%	38,05	1,09	11,58	16,90
RNNB	7	256	256/512	18	71,04	1,71	15,14	41,95	6,11%	5,23%	2,90%	4,74%	58,71	1,31	9,21	23,08

Fonte: Tabela elaborada pela autora

Observa-se que a melhor configuração da rede RNNS possui uma arquitetura de 2 camadas escondidas com 128 e 256 unidades respectivamente. A sequência utilizada foi de 14 meses e foi usada uma camada bidirecional. Nesta configuração, observa-se que o RMSE foi de 9,41 e a MAE de 3,76. Nota-se que o erro RMSE percentual é de 1,23%. Ao comparar as configurações testadas, é possível perceber uma variação do erro RMSE de 78% e MAE de 84% entre os desempenhos.

Tabela 5 – Resultados dos testes *deep smart proxy* com a rede *long short-term memory* (LSTM)

Modelo	Épocas (Early stopping)	Batch Size	Unidades	Sequência	RMSE Petróleo	RMSE Gás	RMSE Água	RMSE Total	%RMSE Petróleo	%RMSE Gás	%RMSE Água	%RMSE Total	MAE Petróleo	MAE Gás	MAE Água	MAE Total
LSTMB	53	256	64/128	18	9,70	0,27	1,80	5,70	0,83%	0,81%	0,35%	0,66%	1,95	0,06	0,75	0,92
LSTMB	48	256	256/512	18	9,71	0,27	1,85	5,71	0,83%	0,83%	0,35%	0,67%	2,27	0,08	0,84	1,06
LSTMB	36	256	128/256	18	10,09	0,28	2,10	5,95	0,87%	0,84%	0,40%	0,70%	2,17	0,06	0,73	0,99
LSTMB	53	256	128/256	14	10,77	0,29	1,88	6,32	0,93%	0,90%	0,36%	0,73%	2,03	0,05	0,95	1,01
LSTMB	38	512	256/1024	14	11,55	0,32	2,56	6,83	0,99%	0,97%	0,49%	0,82%	2,43	0,07	0,89	1,13
LSTMB	50	256	256/512	12	11,85	0,33	1,94	6,93	1,02%	0,99%	0,37%	0,79%	2,30	0,06	0,58	0,98
LSTMB	56	256	128/256	12	12,19	0,34	3,05	7,26	1,05%	1,03%	0,58%	0,89%	2,48	0,08	0,80	1,12
LSTMB	34	256	256/512	14	12,86	0,35	5,99	8,19	1,11%	1,08%	1,15%	1,11%	2,32	0,07	0,98	1,12
LSTM	56	256	128/256	18	35,83	0,98	5,40	20,93	3,08%	3,00%	1,03%	2,37%	11,38	0,31	1,81	4,50
LSTM	51	256	256/512	18	36,66	1,01	5,52	21,41	3,15%	3,07%	1,06%	2,43%	11,52	0,32	2,05	4,63

Fonte: Tabela elaborada pela autora

A melhor configuração da rede LSTM possui uma arquitetura com 2 camadas escondidas de 64 e 128 unidades respectivamente. A sequência utilizada foi de 18 meses e foi usada uma camada bidirecional. Nesta configuração, observa-se que o RMSE foi de 5,70 e a MAE de 0,92. O percentualmente de erro RMSE é de 0,66%. Ao comparar as configurações testadas, é possível perceber uma variação do erro RMSE de 73% e MAE de 80% entre os desempenhos.

Tabela 6 – Resultados dos testes *deep smart proxy* com a Rede *Gated Recurrent Unit* (GRU)

Modelo	Épocas (Early stopping)	Batch Size	Unidades	Sequência	RMSE Petróleo	RMSE Gás	RMSE Água	RMSE Total	%RMSE Petróleo	%RMSE Gás	%RMSE Água	%RMSE Total	MAE Petróleo	MAE Gás	MAE Água	MAE Total
GRUB	51	128	128/256	18	9,42	0,26	1,79	5,54	0,81%	0,79%	0,34%	0,65%	1,86	0,05	0,56	0,82
GRUB	60	256	128/256	18	9,56	0,26	1,87	5,62	0,82%	0,80%	0,36%	0,66%	1,97	0,05	0,53	0,85
GRUB	73	256	64/256	18	9,58	0,27	1,84	5,63	0,82%	0,81%	0,35%	0,66%	1,79	0,06	0,53	0,79
GRUB	53	128	256/512	18	9,61	0,27	2,01	5,67	0,83%	0,81%	0,38%	0,67%	1,92	0,06	0,72	0,90
GRUB	45	64	128/256	18	9,76	0,27	2,07	5,76	0,84%	0,82%	0,40%	0,68%	2,27	0,06	0,69	1,01
GRUB	28	512	256/512	18	10,39	0,29	2,84	6,22	0,89%	0,87%	0,54%	0,77%	2,64	0,07	0,96	1,22
GRUB	43	256	256/512	14	11,12	0,31	2,17	6,54	0,96%	0,94%	0,42%	0,77%	2,39	0,07	1,06	1,17
GRUB	28	256	256/512	12	12,75	0,35	2,60	7,52	1,10%	1,07%	0,50%	0,89%	3,80	0,10	0,80	1,57
GRUB	14	256	256/1024	18	13,07	0,35	4,06	7,90	1,12%	1,07%	0,78%	0,99%	4,99	0,12	1,35	2,15
GRUB	32	256	256/512	10	13,50	0,37	2,43	7,92	1,16%	1,13%	0,47%	0,92%	3,11	0,08	0,71	1,30
GRU	62	256	128/256	18	35,38	0,97	6,00	20,72	3,04%	2,98%	1,15%	2,39%	13,55	0,38	2,55	5,49
GRU	26	128	128/256	18	37,25	1,02	6,56	21,84	3,20%	3,12%	1,26%	2,53%	12,84	0,35	2,82	5,34

Fonte: Tabela elaborada pela autora

Os resultados da Tabela 6 mostram que a melhor configuração da rede GRU possui uma arquitetura com 2 camadas escondidas de 128 e 256 unidades respectivamente. A sequência utilizada foi de 18 meses e foi usada uma camada bidirecional. Nesta configuração observa-se que o RMSE foi de 5,54 e a MAE de 0,82. O percentualmente de erro RMSE total é de 0,65%. Ao comparar as configurações testadas, é possível perceber uma variação do erro RMSE de 75% e MAE de 85% entre os desempenhos.

Diante de todos os resultados do *deep smart proxy*, é possível notar que a rede GRUB gerou o melhor resultado, apresentando uma variação no erro de 87% no RMSE e 96% no MAE, quando comparado com a configuração menos efetiva.

Verifica-se nos modelos *deep smart proxy* que o uso de camadas bidirecionais gerou uma significativa melhora na previsão e redução do erro em comparação com arquiteturas semelhantes que não usaram estas camadas. O modelo GRU apresentou variação de 75% no erro RMSE e 85% no erro MAE enquanto o modelo LSTM apresentou variação de 73% no erro RMSE e 80% no erro MAE. Outro ponto importante é a proximidade dos melhores resultados das redes LSTM e GRU em ambas medidas de erro, mostrando a efetividade das duas redes para previsão. Por outro lado, é possível notar que a rede RNNS não se mostrou tão efetiva para este tipo de previsão devido a sua característica mais simplista, quando comparada aos outros algoritmos de RNR testados.

Se comparadas as redes MLP, as redes *deep learning* do tipo RNR mostraram melhores resultados, provando a efetividade do *deep smart proxy*. O *deep smart proxy* com a rede GRUB apresentou uma redução do erro RMSE de 66% e 79% no erro MAE quando comparado ao *smart proxy*.

## 6

### Previsão com dados observados

#### 6.1

##### Base de dados

Para testar os modelos *smart proxy* e *deep smart proxy*, foi utilizada uma base de dados observados, ou seja, com séries históricas reais de produção de óleo, gás e água. É preciso ressaltar que os dados não correspondem à poços inteligentes, mas o intuito é testar a metodologia usada na proxy para previsão com dados reais de produção.

Neste estudo, a base de dados foi disponibilizada pelo Departamento de Meio Ambiente e Recursos Naturais do estado da Dakota do Sul nos Estados Unidos<sup>4</sup>. A base de dados bruta consiste em 571 séries históricas de produção de diferentes poços e reservatórios. Cada série corresponde à produção de um poço e o seu tamanho varia de acordo com o período em que o poço ficou ativo, ou seja, a partir do momento que foi descoberto até ser definitivamente abandonado. As séries são mensais e compreendem desde o período de 1950 até o ano de 2018.

#### 6.2

##### Mineração de dados (*data mining*)

Foi realizado um trabalho minucioso de mineração de dados antes da realização dos testes. A base bruta continha os seguintes dados: nome do município de localização do poço, número de identificação, nome do poço, coordenada geográfica, estado do poço, mês e ano de produção, número de dias no mês de produção, produção mensal de óleo (medido em barris), produção mensal de gás (medido em mil pés cúbicos ou mcf), produção mensal de água (medido em barris), nome do operador, nome do campo, comentários e indicador de confidencialidade dos dados. Nenhum dado era confidencial.

---

<sup>4</sup> Disponível no site <https://denr.sd.gov/des/og/producti.aspx>. O acesso foi realizado no dia 14 de janeiro de 2019.

Foram desconsideradas as variáveis pouco relevantes para os propósitos desta dissertação, dentre elas: coordenada geográfica, nome do município de localização do poço, nome do operador, nome do campo, comentários e indicador de confidencialidade.

Diferentemente da base de dados sintéticos, nesta base não existe a configuração de válvulas por não se tratar de poços inteligentes. Foi criada uma variável binária para filtrar o estado de produção do poço, onde, “1” seria o poço produzindo e “0” os demais estados, conforme a Tabela 7 abaixo.

Tabela 7 – Situação do poço e variável binária correspondente

<b>Estado do poço</b>	<b>Variável binária correspondente</b>
Injeção de gás	0
Poço de abastecimento de água	0
Injetor	0
Não reportado	0
Conectado e abandonado	0
Produzindo	1
Reentrada	0
Poço de monitoramento do reservatório	0
Eliminação de água	0
Fechado	0
Temporariamente abandonado	0
Injeção de água	0
Injeção de água - reentrada	0
Poço de injeção de água	0

Fonte: Tabela elaborada pela autora

Com a criação da variável binária, a coluna correspondente do estado do poço na Tabela 7 foi desconsiderada. Em outras palavras, somente a coluna correspondente com a variável binária serviu de entrada para o aprendizado da rede. O intuito é que esta variável funcione como a válvula geral do poço, de forma que somente quando aberta e igual a “1” estaria produzindo.

O número de identificação, data e nome do poço foram mantidos somente para ordenamento da base de dados e foram desconsiderados para o aprendizado da rede. Cada linha da base de dados corresponde a uma série histórica de um poço. A data exata em si não importa para o aprendizado da rede, pois o que importa é manter intacta a ordem/sequência da produção. As variáveis de cada série ficaram lado a lado, ou seja, na mesma linha, mas em diferentes colunas ordenadas no tempo (estado do poço e produção de óleo, gás e água no tempo).

Uma vez que os dias de produção e o respectivo total mensal produzido estavam com grande variação mesmo dentro da uma mesma série histórica, foram feitas as divisões da quantidade produzida de óleo, água e gás pelo número de dias produzidos. Desta forma, obteve-se a média diária de produção em cada mês. A justificativa para isso é que a previsão se torna mais precisa por não ter que considerar quantos dias ativos o poço terá por mês, principalmente pelo fato dele poder se tornar se tornar “inativo” por diversas razões. O importante, neste caso, é obter uma previsão de produção para o dia que o poço estará ativo e não prever quantos dias ele estará ativo. Para o aprendizado da rede, portanto, considerou-se apenas a média diária de produção de óleo, gás e água. A previsão da média diária de produção, e não do total do mês, foi realizada.

Foram desconsideradas as séries que não possuíam dados suficientes de produção (neste caso a rede não teria o que aprender) ou eram muito curtas (menor que 60 meses). Também foram excluídos poços que só produziam gás e séries em que o poço ficou a maior parte do tempo não produzindo. Deste modo, das 571 séries históricas disponíveis, foram consideradas 316 após a mineração de dados.

Por conter diferentes variáveis com diferentes medidas para estimativa de cada produção, a base de dados foi normalizada antes do treinamento, validação e testes.

### 6.3

#### Testes e Resultados

O objetivo do presente capítulo é testar os algoritmos de *machine learning* aplicados nos modelos *smart proxy* e *deep smart proxy* com uma base de dados observados de produção. Os algoritmos testados foram: MLP, RNNS, GRU e LSTM.

Assim como os testes anteriores, todos os testes com a base de dados real também foram realizados usando Python na versão 3.6 e no cluster do Intel AI Dev Cloud. A métrica para estimativa do melhor desempenho foi pelo valor do menor RMSE total.

Visando evitar o problema de *overfitting*, para todos os testes foram utilizados a metodologia de parada antecipada ou *early stopping*. No treinamento, o erro quadrático médio (MSE) foi utilizado como função para medir a perda. Os dados foram divididos em conjuntos de treinamento, validação e testes, sendo 80% para treinamento e validação e 20% para teste.

Para os testes usando MLP, os pesos foram inicializados de forma aleatória e os demais parâmetros básicos foram os mesmos apresentados na Tabela 2. Todos os resultados foram apresentados por ordem de melhor desempenho, sendo, portanto, o melhor resultado na primeira posição em destaque. A métrica para determinar o melhor desempenho foi dada pelo RMSE total. Os resultados dos testes da rede MLP se encontram na Tabela 8 abaixo.

Tabela 8 – Resultados dos testes com dados observados usando Redes Neurais Multilayer Perceptron (MLP)

Modelo	Épocas (Early stopping)	Batch Size	Neurônios	Sequência	RMSE Petróleo	RMSE Gás	RMSE Água	RMSE Total	%RMSE Petróleo	%RMSE Gás	%RMSE Água	%RMSE Total	MAE Petróleo	MAE Gás	MAE Água	MAE Total
MLP	13	64	32/64	14	6,97	57,45	45,58	42,53	1,29%	0,64%	0,77%	0,90%	3,30	21,89	16,31	13,83
MLP	17	32	16/32	10	8,04	73,18	34,19	46,87	1,49%	0,82%	0,57%	0,96%	3,80	29,50	15,75	16,35
MLP	10	64	32/64	10	7,08	71,09	41,84	47,80	1,30%	0,79%	0,70%	0,93%	3,35	24,48	20,69	16,17
MLP	8	64	64/128	18	8,34	71,99	46,05	49,59	1,55%	0,80%	0,77%	1,04%	4,69	35,15	19,89	19,91
MLP	8	64	32/64	12	7,57	75,82	43,97	50,97	1,40%	0,85%	0,74%	1,00%	4,31	44,89	13,56	20,92
MLP	33	32	32/64	14	7,27	74,16	71,73	59,72	1,35%	0,83%	1,20%	1,13%	4,40	31,52	13,93	16,62
MLP	12	64	32/64	18	7,40	105,56	38,08	64,93	1,37%	1,18%	0,64%	1,06%	4,07	41,26	19,74	21,69
MLP	23	32	16/32	14	7,46	92,03	74,84	68,62	1,38%	1,03%	1,26%	1,22%	3,42	32,38	18,52	18,11
MLP	7	64	64/128	12	9,76	105,79	77,75	76,01	1,81%	1,18%	1,31%	1,43%	5,42	30,43	30,10	21,98
MLP	8	32	128/246	14	8,38	101,56	99,20	82,11	1,55%	1,13%	1,67%	1,45%	5,44	32,66	19,94	19,34

Fonte: Tabela elaborada pela autora

Nota-se que a melhor configuração da rede MLP possui uma arquitetura com 2 camadas escondidas de 32 e 64 neurônios respectivamente. A sequência utilizada foi de 14 meses. Nesta configuração, observa-se que o RMSE foi de 45,58 e a MAE de 13,83. Percentualmente o erro RMSE total foi de 0,9%. Ao comparar

as configurações testadas, é possível perceber uma variação do erro RMSE de 48% e MAE de 28% entre os desempenhos.

Para os testes usando algoritmos de *deep learning* do tipo RNR, o método de otimização utilizado foi o Adam. Assim como nos testes realizados no *deep smart proxy*, foram realizados testes com camadas bidirecionais. Nestes, a letra “B” foi destacada. Ela se encontra na coluna com os métodos de *machine learning* (ex. GRUB, LSTMB e RNNsB). Os resultados da rede RNNs foram ilustrados na Tabela 9, abaixo.

Tabela 9 – Resultados dos testes com dados observados usando a Rede Neural Recorrente Simples (RNNs)

Modelo	Épocas (Early stopping)	Batch Size	Unidades	Sequência	RMSE Petróleo	RMSE Gás	RMSE Água	RMSE Total	%RMSE Petróleo	%RMSE Gás	%RMSE Água	%RMSE Total	MAE Petróleo	MAE Gás	MAE Água	MAE Total
RNNB	25	128	256/512	14	4,32	57,98	34,31	38,98	0,80%	0,65%	0,56%	0,67%	2,30	21,68	19,71	14,56
RNNB	28	128	256/512	18	5,17	64,14	28,84	40,71	0,96%	0,72%	0,47%	0,72%	3,55	25,54	13,06	14,05
RNNB	32	256	128/256	12	4,99	69,27	29,10	43,47	0,92%	0,77%	0,48%	0,73%	3,30	39,11	19,19	20,53
RNNB	17	128	256/512	12	4,06	69,52	32,19	44,29	0,75%	0,78%	0,53%	0,69%	1,90	45,30	22,45	23,22
RNNB	25	256	128/512	10	4,07	74,64	56,65	54,15	0,75%	0,83%	0,93%	0,84%	1,82	44,30	44,02	30,04
RNNB	21	256	128/256	18	5,28	80,98	50,69	55,25	0,98%	0,90%	0,83%	0,91%	3,18	49,05	39,08	30,44
RNN	21	256	128/256	12	6,55	115,22	50,45	72,72	1,21%	1,29%	0,83%	1,11%	3,21	44,71	29,75	25,89
RNNB	9	256	64/128	12	14,18	123,20	56,81	78,76	2,63%	1,38%	0,93%	1,64%	13,00	70,61	25,95	36,52
RNN	13	256	128/256	18	6,99	112,46	79,66	79,67	1,30%	1,26%	1,31%	1,29%	3,52	49,30	63,79	38,87
RNN	13	128	256/512	12	7,74	120,24	70,16	80,50	1,43%	1,34%	1,15%	1,31%	4,78	45,27	46,55	32,20

Fonte: Tabela elaborada pela autora

Os resultados da Tabela 9 mostram que a melhor configuração da rede RNNs possui uma arquitetura com 2 camadas escondidas de 256 e 512 unidades respectivamente. A sequência utilizada foi de 14 meses e foi usada uma camada bidirecional. Nesta configuração, observa-se que o RMSE foi de 38,98 e a MAE de 14,56. Percentualmente o erro RMSE total foi de 0,67%. Ao comparar as configurações testadas, é possível perceber uma variação do erro RMSE de 52% e MAE de 55% entre os desempenhos.

Para a rede LSTM, os resultados dos testes podem ser observados na Tabela 10.

Tabela 10 – Resultados dos testes com dados observados usando a *Rede long short-term memory* (LSTM)

Modelo	Épocas (Early stopping)	Batch Size	Unidades	Sequência	RMSE Petróleo	RMSE Gás	RMSE Água	RMSE Total	%RMSE Petróleo	%RMSE Gás	%RMSE Água	%RMSE Total	MAE Petróleo	MAE Gás	MAE Água	MAE Total
LSTMB	48	256	128/256	12	3,91	53,75	18,05	32,81	0,73%	0,60%	0,30%	0,54%	1,77	13,87	4,62	6,75
LSTMB	54	256	128/256	18	3,87	53,64	19,88	33,10	0,72%	0,60%	0,33%	0,55%	1,62	13,37	9,56	8,18
LSTMB	40	256	128/512	12	3,50	53,96	20,36	33,36	0,65%	0,60%	0,33%	0,53%	0,95	17,16	10,11	9,41
LSTMB	31	256	128/512	10	3,51	55,49	19,08	33,94	0,65%	0,62%	0,31%	0,53%	0,86	18,02	7,07	8,65
LSTMB	31	256	128/1024	12	3,50	56,73	19,14	34,63	0,65%	0,63%	0,31%	0,53%	1,07	22,75	7,21	10,34
LSTMB	31	64	128/1024	12	3,58	53,99	26,66	34,83	0,66%	0,60%	0,44%	0,57%	1,20	19,17	9,90	10,09
LSTMB	24	256	256/512	18	3,73	76,21	31,77	47,72	0,69%	0,85%	0,52%	0,69%	1,25	26,99	14,13	14,12
LSTM	90	128	128/256	12	6,09	89,32	37,54	56,05	1,13%	1,00%	0,62%	0,91%	2,52	22,74	17,44	14,23
LSTM	274	256	128/512	10	6,05	92,34	34,56	57,03	1,12%	1,03%	0,57%	0,91%	2,54	20,07	10,51	11,04
LSTM	42	256	128/256	18	6,51	92,24	39,29	58,06	1,21%	1,03%	0,64%	0,96%	3,24	27,39	15,14	15,26
LSTM	33	256	128/512	12	6,35	98,91	41,66	62,07	1,18%	1,10%	0,68%	0,99%	2,89	29,20	22,64	18,25

Fonte: Tabela elaborada pela autora

Nota-se que a melhor configuração da rede LSTM possui uma arquitetura com 2 camadas escondidas de 128 e 512 unidades respectivamente. A sequência utilizada foi de 12 meses e foi usada uma camada bidirecional. Nesta configuração, observa-se que o RMSE foi de 32,81 e a MAE de 6,75. Percentualmente, o erro RMSE total foi de 0,54%. Ao comparar as configurações testadas, é possível perceber uma variação do erro RMSE de 47% e MAE de 63% entre os desempenhos.

Os resultados dos testes com a rede GRU podem ser observados na Tabela 11.

Tabela11 – Resultados dos testes com dados observados usando a Rede *Gated Recurrent Unit* (GRU)

Modelo	Épocas (Early stopping)	Batch Size	Unidades	Sequência	RMSE Petróleo	RMSE Gás	RMSE Água	RMSE Total	%RMSE Petróleo	%RMSE Gás	%RMSE Água	%RMSE Total	MAE Petróleo	MAE Gás	MAE Água	MAE Total
GRUB	28	256	128/512	12	3,75	52,29	18,79	32,15	0,69%	0,58%	0,31%	0,53%	1,78	10,00	6,34	6,04
GRUB	31	256	128/1024	12	3,63	53,36	18,34	32,66	0,67%	0,60%	0,30%	0,52%	1,43	14,69	4,06	6,73
GRUB	36	256	128/256	12	3,48	52,88	21,48	33,01	0,65%	0,59%	0,35%	0,53%	0,76	13,92	12,92	9,20
GRUB	27	256	128/256	18	4,17	53,23	20,67	33,06	0,77%	0,59%	0,34%	0,57%	2,32	10,20	10,19	7,57
GRUB	28	256	128/512	10	3,53	54,95	19,88	33,80	0,65%	0,61%	0,33%	0,53%	0,97	18,58	9,78	9,78
GRUB	33	256	128/1024	10	4,54	62,78	32,87	41,00	0,84%	0,70%	0,54%	0,69%	3,13	35,47	26,53	21,71
GRUB	28	128	128/512	12	3,99	72,30	27,24	44,67	0,74%	0,81%	0,45%	0,66%	2,11	48,74	17,84	22,90
GRUB	11	64	128/1024	12	5,02	65,48	49,27	47,40	0,93%	0,73%	0,81%	0,82%	1,98	26,96	24,99	17,98
GRU	21	256	128/256	18	6,81	96,27	40,52	60,43	1,26%	1,07%	0,67%	1,00%	3,78	32,55	17,19	17,84
GRU	29	256	128/256	10	6,60	102,04	37,66	62,92	1,22%	1,14%	0,62%	0,99%	3,52	23,92	13,53	13,66
GRU	17	256	128/512	12	6,42	106,22	40,56	65,75	1,19%	1,19%	0,67%	1,01%	3,05	38,63	17,90	19,86
GRU	7	64	128/1024	12	6,49	112,52	43,09	69,66	1,20%	1,26%	0,71%	1,05%	3,37	40,38	22,23	21,99

Fonte: Tabela elaborada pela autora

É possível observar nos resultados encontrados na rede GRU que o melhor desempenho apresentou o RMSE com o valor de 32,15 e MAE de 6,04. Percentualmente, o erro RMSE total foi de 0,53%. Vale ressaltar que, na sua arquitetura, foram usadas camadas bidirecionais com uma arquitetura de 2 camadas escondidas com 128 e 512 unidades respectivamente. A sequência utilizada foi de 12 meses. Ao comparar as configurações testadas, é possível perceber uma variação do erro RMSE de 54% e MAE de 73% entre os desempenhos.

Diante de todos os resultados com o uso de algoritmos *deep learning* do tipo RNR, é possível notar que a rede GRUB gerou o melhor resultado, apresentando uma variação no erro de 60% no RMSE e 81% no MAE quando comparado à configuração de outras RNR com menor desempenho. Verificou-se que, nos modelos *deep learning* do tipo RNR, o uso de camadas bidirecionais gerou uma significativa melhora na previsão e redução do erro ao comparar com arquiteturas semelhantes que não usaram estas camadas, chegando a uma variação de 54% no erro RMSE e 3% no erro MAE para os modelos GRU e a variação de 47% no erro RMSE e 63% no erro MAE no modelo LSTM. Outro ponto importante é a proximidade dos melhores resultados das redes LSTM e GRU em ambas medidas de erro, mostrando a efetividade das duas redes para previsão. Por outro lado, é possível notar que a rede RNNs não se mostrou tão efetiva quantos as demais redes

*deep learning* testadas para este tipo de previsão, devido a sua característica mais simplista.

Em comparação com as redes MLP, a rede GRUB apresentou uma redução do erro RMSE de 24% e 53% no erro MAE. Outro ponto importante é no tangente ao número de meses (sequência). Neste quesito, o melhor desempenho da MLP usou 14 meses, comparado aos 12 meses do GRUB.

Conclui-se desta forma que a rede GRU teve o melhor resultado com menor erro em ambas as medidas RMSE e MAE. Vale ressaltar que o uso de camadas bidirecionais gerou uma significativa melhora na redução do erro. Outro ponto importante é que os melhores resultados das redes GRU e LSTM (pelos quais usaram camadas bidirecionais) apresentaram valores bem próximos de RMSE total e MAE, mostrando por esta métrica, a efetividade das duas redes para previsão.

## Conclusão

Este trabalho teve por objetivo explorar as vantagens dos algoritmos de *machine learning*, em especial do algoritmo de *deep learning* do tipo RNR, para realizar a previsão de produção de óleo, gás e água. Para tanto, partiu-se dos dados sintéticos fornecidos pelo sistema de poços inteligentes usando a metodologia de *data driven*. Foi realizado um comparativo do desempenho de diferentes algoritmos de *machine learning* como as redes neurais convencionais (*multilayer perceptron*), redes neurais recorrentes simples, LSTM, GRU e camadas bidirecionais.

Para o modelo com base de dados sintéticos, em comparação com as redes MLP, as redes *deep learning* do tipo RNR mostraram melhores resultados, provando a efetividade do *deep smart proxy*. O *deep smart proxy* com a rede GRUB apresentou uma redução do erro RMSE de 66% e 79% no erro MAE quando comparado ao *smart proxy*. Outro ponto importante é a magnitude do erro, que percentualmente no GRUB representou 0,65%.

Para o modelo com base de dados observados, na comparação com as redes MLP, a rede GRUB apresentou uma redução do erro RMSE de 24% e 56% no erro MAE. Outro ponto importante é o número de meses (sequência), fator no qual a MLP apresentou melhor desempenho usando 14 meses comparados a 12 meses do GRU.

O uso de camadas bidirecionais gerou uma redução do erro se comparadas a arquiteturas semelhantes que não usaram estas camadas em ambas as bases de dados. Esta variação na rede GRU chegou a 75% no erro RMSE e 85% no erro MAE na base com dados sintéticos e 54% no erro RMSE e 73% no erro MAE na base de dados observados. Percentualmente, o erro RMSE total foi bem próximo em ambas as bases de dados, sendo na rede GRUB de 0,53% nos dados observados e 0,65% nos dados sintéticos.

Desta forma, conclui-se que, em ambas as bases de dados, os testes com *deep learning* do tipo RNR apresentaram melhores resultados em comparação com os modelos de redes neurais convencionais (MLP) para previsão, principalmente com o uso da arquitetura bidirecional.

A rede GRUB (com arquitetura bidirecional) apresentou o menor RMSE para prever a produção de óleo, gás e água, tanto ao se utilizar a base de dados simulados, quanto a de dados observados. Convém mencionar que, apesar da rede GRUB ter tido o melhor desempenho, a rede LSTMB (com arquitetura bidirecional) também apresentou resultados bastante próximos.

No que diz respeito ao esforço computacional, a rede GRU por ser mais simples do que a rede LSTM, pode ser treinada com mais rapidez e, no caso específico deste trabalho, mais eficiente em sua execução. No entanto, a LSTM pode ser mais expressiva e, com mais dados e parâmetros, pode levar a resultados melhores. Em geral, o tempo computacional foi menor nos testes com a *smart proxy* e MLP na base de dados observados. No entanto, em ambos os casos o RMSE foi significativamente maior.

## Sugestões para futuros trabalhos

Como extensão da presente dissertação, há a sugestão de melhorias que busquem otimizar o processamento de simulações de reservatórios, que são computacionalmente intensivas, além de reduzir o erro no processo de previsão de produção de petróleo.

A fim de reduzir o tempo de simulação, recomenda-se a aplicação de algoritmos como as Redes Geradoras Adversariais ou *Generative Adversary Networks* (GAN) para a geração de curvas de produção e alternativas de controle de válvulas. Tais algoritmos são capazes de extrair representações de alto nível para gerar novas amostras que seguem a mesma função de distribuição probabilística de um respectivo conjunto de dados de treinamento. Esta metodologia irá permitir o aumento do conjunto de dados de forma precisa o suficiente para o treinamento da rede, diminuindo a necessidade de simulações.

No tangente à redução do erro no processo de previsão de produção, recomenda-se o teste com diferentes algoritmos de *deep learning* e metodologias híbridas. Além disso, sugere-se a aplicação da mesma metodologia aplicada no presente trabalho em diferentes modelos de reservatórios e maior quantidade variáveis como, por exemplo, maior número de poços, mais zonas de produção, maior quantidade válvulas e simulações de diferentes estados geológicos.

Por fim, sugere-se a criação de modelos que incorporem variáveis econômicas (como o Valor Presente Líquido) e incertezas.

## Referências bibliográficas

ABREU, ANA CAROLINA ALVES. **Uma abordagem para avaliação de flexibilidade considerando incerteza e informação futura: uma aplicação para poços inteligentes**. Tese de Doutorado – Departamento de Engenharia Elétrica, Pontifícia Universidade Católica do Rio de Janeiro. Rio de Janeiro, 2016.

ABREU, A.C.; M.A.G. DIAS; M.A.C. PACHECO; A.A. EMERICK & F. BORGES. **FlexWell: smart wells flexibility management and valuation under uncertainty**. Artigo apresentado na 21st Annual International Real Options Conference, Boston, 2017.

AL-FATTAH, S. M., & STARTZMAN, R. A. **Neural Network Approach Predicts U.S. Natural Gas Production**. Society of Petroleum Engineers. doi:10.2118/67260-MS, 2001.

AL-OTAIBI, B. Z.; SCHECHTER, D. S. & WATTENBARGER, R. A. **Production Forecast, Analysis and Simulation of Eagle Ford Shale Oil Wells**. Society of Petroleum Engineers. doi:10.2118/172929-MS, 2015.

ALENEZI, F., & MOHAGHEGH, S. **Developing a Smart Proxy for the SACROC Water-Flooding Numerical Reservoir Simulation Model**. Society of Petroleum Engineers. doi:10.2118/185691-MS, 2017.

ALTHELAYA, K. A.; EL-ALFY, E. M. & MOHAMMED, S. **Evaluation of bidirectional LSTM for short-and long-term stock market prediction**. 9th International Conference on Information and Communication Systems (ICICS), Irbid, 2018, pp. 151-156. doi: 10.1109/IACS.2018.8355458, 2018.

BALAJI, K.; RABIEI, M.; SUICMEZ, V.; CANBAZ, C. H.; AGHARZEYVA, Z.; TEK, S.; TEMIZEL, C. **Status of Data driven Methods and their Applications in Oil and Gas Industry**. Society of Petroleum Engineers. doi:10.2118/190812-MS, 2018.

BOOMER, ROBERT J. **Predicting Production Using a Neural Network (Artificial Intelligence Beats Human Intelligence)**. Society of Petroleum Engineers. doi:10.2118/30202-MS, 1995.

BURDA, B.; CROMPTON, J.; SARDOFF, H. & FALCONER, J. **Information Architecture Strategy for the Digital Oil Field**. Paper SPE-106687-MS, presented at the 2007 SPE Digital Energy Conference and Exhibition, Houston, 2007.

CAO, Q.; BANERJEE, R.; GUPTA, S.; LI, J.; ZHOU, W.; & JEYACHANDRA, B. **Data driven Production Forecasting Using Machine Learning**. Paper SPE 180984-MS, presented at the SPE Argentina Exploration and Production of Unconventional Resources Symposium, Buenos Aires, 2016.

CHO, KYUNGHYUN; VAN MERRIENBOER, BART; GULCEHRE, CAGLAR; BAHDANAU, DZMITRY; BOUGARES, FETHI; SCHWENK, HOLGER; BENGIO, YOSHUA. **Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation**. arXiv:1406.1078, 2014.

CRAMER, R.; GOH, K.-C. **Data driven Surveillance and Optimization for Gas, Subsea and Multizone Wells**. Society of Petroleum Engineers. doi:10.2118/122554-MS, 2009.

CRNKOVIC-FRIIS, L. & ERLANDSON, M. **Geology Driven EUR Prediction Using Deep Learning**. Society of Petroleum Engineers. doi:10.2118/174799-MS, 2015.

DURLOFSKY, LOUIS J.; AZIZ, KHALID. **Optimization of Smart Well Control**. Society of Petroleum Engineers. doi:10.2118/79031-MS, 2002.

GAO, C. H.; RAJESWARAN, R. T. & NAKAGAWA, E. Y. **A Literature Review on Smart Well Technology**. Society of Petroleum Engineers. doi:10.2118/106011-MS, 2007.

GLANDT, CARLOS A. **Reservoir Aspects of Smart Wells**. Society of Petroleum Engineers. doi:10.2118/81107-MS, 2003.

GOH, K.-C.; MONCUR, C. E.; VAN OVERSCHEE, P. & BRIERS, J. **Production Surveillance And Optimization With Data driven Models**. International Petroleum Technology Conference. doi:10.2523/IPTC-11647-MS, 2007.

GOH, K.-C.; DALE-PINE, B.; YONG, I. H. W.; VANOVERSCHEE, P. & LAUWERYS, C. **Production Surveillance and Optimization for Multizone Smart Wells with Data driven Models**. Society of Petroleum Engineers. doi:10.2118/112204-MS, 2008.

GOODFELLOW, I.; BENGIO, Y. & COURVILLE, A. **Deep Learning**. MIT Press, 2016. Disponível em: < <http://www.deeplearningbook.org>>. Acesso em: 25 de outubro de 2018.

HAYKIN, SIMON. **Redes Neurais: Princípios e prática**. Tradução Paulo Martins Engel – 2 ed., Porto Alegre, Bookman, 2007.

HE, Q.; MOHAGHEGH, S. D. & LIU, Z. **Reservoir Simulation Using Smart Proxy in SACROC Unit - Case Study**. Society of Petroleum Engineers. doi:10.2118/184069-MS, 2016.

HOCHREITER, SEPP & SCHMIDHUBER, JURGEN. **Long Short-Term Memory**, Neural Computation, 1997.

HOLDAWAY, K.R. **Harness Oil and Gas Big Data with Analytics – Optimize Exploration and Production with Data driven Models**. John Wiley & Sons, Inc., Hoboken (USA), 2014.

HOLDAWAY, K. R.; LAING, M. L. **Drilling and Completion Optimization in Unconventional Reservoirs with Data driven Models**. Society of Petroleum Engineers. doi:10.2118/176820-MS, 2015.

HOLDAWAY, K.R. & IRVING, D.H.B. **Enhance Oil & Gas Exploration with Data driven Geophysical and Petrophysical Models**. John Wiley & Sons, Inc., Hoboken (USA), 2018.

JACOBS, T. **Devon Energy Rises to the Top as a Data driven Producer**. Journal of Petroleum Technology, 2016.

JAFARIZADEH, B. & BRATVOLD, R.B. **Strategic Decision Making in the Digital Oil Field**. Paper SPE-123213-MS-P, presented at the 2009 SPE Digital Energy Conference and Exhibition, Houston, 2009.

JAJA, A. J.; OGUNTONA, B. & EME, V. **Improved Short Term Oil Production Forecast – Application of Well Decline Rates to Generate Field Decline Rate**. Society of Petroleum Engineers. doi:10.2118/184251-MS, 2016.

JIA, X. & ZHANG, F. **Applying Data driven Method to Production Decline Analysis and Forecasting**. Paper SPE-181616-MS, presented at the SPE Annual Technical Conference and Exhibition, Dubai, 2016.

JING, S.; ERROUISSI, R.; AL-DURRA, A. & BOIKO, I. **A data driven subspace predictive controller design for artificial gas-lift process**. 2015 IEEE Conference on Control Applications (CCA), Sydney, NSW, pp. 1179-1184. doi: 10.1109/CCA.2015.7320772, 2015.

KINGMA, DIEDERIK P. & BA, JIMMY LEI. **Adam: a method for stochastic optimization**. International Conference on Learning Representations, 2015. Disponível em: < <https://arxiv.org/pdf/1412.6980.pdf> >. Acesso em: 12 de março de 2019.

KLIE, H. **Physics-Based and Data driven Surrogates for Production Forecasting**. Paper SPE 173206-MS, presented at the SPE Reservoir Simulation Symposium, Houston, 2015.

KOHLER, MANOELA RABELLO. **Redes Neurais Aplicadas à Construção de Aproximadores para Simulação Integrada entre Reservatório e Sistema de Produção**. Dissertação de Mestrado, Pontifícia Universidade Católica do Rio de Janeiro, 2013.

LEI SUN, YANGE BI & LU, GUORONG. **Application of BP Neural Network in Oil Field Production Prediction**. Second World Congress on Software Engineering, Wuhan, 2010, pp. 201-203. doi: 10.1109/WCSE.2010.101, 2010.

MAUCEC, M., & GARNI, S. **Application of Automated Machine Learning for Multi-Variate Prediction of Well Production**. Society of Petroleum Engineers. doi:10.2118/195022-MS, 2019.

MOHAGHEGH, S. D.; ABDULLA, F.; ABDOU, M.; GASKARI, R. & MAYSAM, M. **Smart Proxy: An Innovative Reservoir Management Tool; Case Study of a Giant Mature Oilfield in the UAE.** Society of Petroleum Engineers. doi:10.2118/177829-MS, 2015.

MOISÉS, G.V.; ROLIM, T.A. & FORMIGLI, J.M. **GeDIg – Petrobras Corporate Program for Digital Integrated Field Management.** Paper SPE-112153-MS-P, Artigo apresentado em 2008 SPE Intelligence Energy Conference and Exhibition, Amsterdam, 2008.

MONTEIRO, D. D.; FERREIRA-FILHO, V. M.; CHAVES, G. S.; DE SANTANA, R. S.; DUQUE, M. M.; GRANJA-SAAVEDRA, A. L.; BAIOCO, J.S.; VIEIRA, B.F.; TEIXEIRA, A. F. **Uncertainty Analysis for Production Forecast in Oil Wells.** Society of Petroleum Engineers. doi:10.2118/185550-MS, 2017.

OLAH, CHRISTOPHER. **Understanding LSTM networks.** Recurrent Neural Networks, 2015. Disponível em: < <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>> Acesso em: 10 de maio de 2019.

RAJVANSHI, A. K.; GMELIG MEYLING, R., & TEN HAAF, D. **Instilling Realism in Production Forecasts: Dos and Don'ts.** Society of Petroleum Engineers. doi:10.2118/155443-MS, 2012.

RANJITH, R.; SUHAG, A.; BALAJI, K.; PUTRA, D.; DHANNOON, D.; SARACOGLU, O.; AMINZADEH, F. **Production Optimization Through Utilization of Smart Wells in Intelligent Fields.** Society of Petroleum Engineers. doi:10.2118/185709-MS, 2017.

RUSSO, E.E.R.; AMOROSO, J.C. & ROLIM, T.A. **Benefits Quantification and Change Management in Petrobras Corporate Program for Digital Integrated Field Management (GeDIg).** Paper SPE-128675-MS, presented at the 2010 SPE Intelligence Energy Conference and Exhibition, Utrecht, 2010.

SCHUSTER, MIKE & PALIWAL, KULDIP. K. **Bidirectional recurrent neural networks.** IEEE Transactions on Signal Processing, vol. 45, no. 11, pp. 2673-2681, doi: 10.1109/78.650093, Nov. 1997.

STONE, P. **Introducing Predictive Analytics: Opportunities.** Paper SPE 106865, presented at the 2007 SPE Digital Energy Conference and Exhibition, Houston, 2007.

SUN, J.; MA, X. & KAZI, M. **Comparison of Decline Curve Analysis DCA with Recursive Neural Networks RNN for Production Forecast of Multiple Wells.** Society of Petroleum Engineers. doi:10.2118/190104-MS, 2018.

TALAVERA, A. L.; TÚPAC, Y. J.; VELLASCO, M. M. B. R. **Controlling Oil Production in Smart Wells by MPC Strategy With Reinforcement Learning.** Society of Petroleum Engineers. doi:10.2118/139299-MS, 2010.

TEMIZEL, CENK.; NABIZADEH, MEHDI.; KADKHODAEI, NEMATOLLAH.; RANJITH, RAHUL; SUHAG, ANUJ; BALAJI, KARTHIK; DHANNOON, DIYAR. **Data driven Optimization of Injection-Production in Waterflood**

**Operations.** Paper SPE 187468-MS, presented at the Intelligent Oil and Gas Symposium, Abu Dhabi, 2017.

TEMIZEL, CENK; PUTRA, DIKE; ANGGRAINI, HENNY; MORENO, RAUL; DROZDOV, OLEG; SKIBA, ANDREY. **Effective Use of Data driven Methods in Brown Fields.** Society of Petroleum Engineers. doi:10.2118/187840-MS, 2017.

WEISS, WILLIAM W.; BALCH, ROBERT S., & STUBBS, BRUCE. A. **How Artificial Intelligence Methods Can Forecast Oil Production.** Society of Petroleum Engineers. doi:10.2118/75143-MS, 2002.

WONG, A. K. C.; WANG, YANG. **Pattern discovery: a data driven approach to decision support.** IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol.33, no.1, pp.114-124, doi: 10.1109/TSMCC.2003.809869, 2003.

ZANGL, G.; GRAF, T., & AL-KINANI, A. **Proxy Modeling in Production Optimization.** Society of Petroleum Engineers. doi:10.2118/100131-MS, 2006.

ZHANG, J.; CHEUNG, S. W.; EFENDIEV, Y.; GILDIN, E. & CHUNG, E. T. **Deep Model Reduction-Model Learning for Reservoir Simulation.** Society of Petroleum Engineers. doi:10.2118/193912-MS, 2019.

ZHANG, XU; ZOU, YUANYUAN; LI, SHAOYUAN & XU, SHENGHU. **Product Yields Forecasting for FCCU via Deep Bi-directional LSTM Network.** 2018 37th Chinese Control Conference (CCC), Wuhan, pp. 8013-8018. Doi: 10.23919/ChiCC.2018.8483420, 2018.