

UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

ANTONIO CARLOS GIMENES JUNIOR

MODELAGEM E SIMULAÇÃO DA AUTODEPURAÇÃO EM RIOS

DISSERTAÇÃO DE MESTRADO

MEDIANEIRA

2023

ANTONIO CARLOS GIMENES JUNIOR

MODELAGEM E SIMULAÇÃO DA AUTODEPURAÇÃO EM RIOS

Modeling And Simulation Of Self-Purification In Rivers

Dissertação apresentada como requisito para obtenção do título de Mestre em Tecnologias Computacionais para o Agronegócio da Universidade Tecnológica Federal do Paraná (UTFPR).

Orientador(a): Prof^a. Dr^a. Fabiana Costa Araújo Schütz

MEDIANEIRA

2023



[4.0 Internacional](https://creativecommons.org/licenses/by-nc-sa/4.0/)

Esta licença permite remixe, adaptação e criação a partir do trabalho, para fins não comerciais, desde que sejam atribuídos créditos ao(s) autor(es) e que licenciem as novas criações sob termos idênticos. Conteúdos elaborados por terceiros, citados e referenciados nesta obra não são cobertos pela licença.



ANTONIO CARLOS GIMENES JUNIOR

MODELAGEM E SIMULAÇÃO DA AUTODEPURAÇÃO EM RIOS

Trabalho de pesquisa de mestrado apresentado no dia como requisito para a obtenção do título de Mestre em Tecnologias Computacionais para o Agronegócio, da Universidade Tecnológica Federal do Paraná (UTFPR). Área de concentração: Tecnologias Computacionais Aplicadas À Produção Agrícola E Agroindústria.

Data de aprovação: 25 de fevereiro 2021

Profa. Dra. Fabiana Costa De Araujo Schutz - Universidade Tecnológica Federal do Paraná

Prof. Dr. Andre Sandmann - Universidade Tecnológica Federal do Paraná

Prof. Dr. Paulo Lopes De Menezes - Universidade Tecnológica Federal do Paraná

Prof. Dr. Paulo Sergio Graziano Magalhaes - Universidade Estadual de Campinas

Documento gerado pelo sistema acadêmico da UTFPR a partir dos dados da Ata de defesa em 26/02/2021

RESUMO

O presente trabalho descreve o desenvolvimento de um modelo em Redes Neurais Artificiais (RNA) com o objetivo de simular a capacidade de um rio de retomar a quantidade de oxigênio dissolvido (OD), após ter recebido uma carga poluidora. O experimento foi desenvolvido em um rio localizado no município de Medianeira no Estado do Paraná que recebe o efluente tratado de uma agroindústria frigorífica. O modelo foi desenvolvido com base em dados da qualidade da água do rio e do efluente que é incorporado ao curso de água ao longo do intervalo estudado. Para treinamento e validação do modelo, foram utilizados dados de estudos anteriores, coletados em 2014. As variáveis de entrada que foram usadas na rede são os parâmetros de qualidade da água e do efluente e a saída foi o OD. Para tanto, foram desenvolvidos cinco modelos de RNA, que sofreram modificações nas variáveis de entrada, porém mantiveram a mesma saída e comparou-se os resultados simulados pela RNA com os resultados coletados *in loco*. Além disso, os resultados obtidos, foram comparados com os resultados de estudo realizado em 2014, com o objetivo de se verificar a evolução das tecnologias utilizadas em RNA. Dentre os principais resultados do presente estudo destaca-se a velocidade de processamento em relação ao estudo anterior. Dentre os modelos desenvolvidos, o modelo nomeado RNA3 foi o que apresentou resultados mais similares aos resultados coletados em campo, demonstrando maior capacidade de simulação da autodepuração do rio.

Palavras-chave: qualidade da água; DBO; OD; redes neurais.

ABSTRACT

The present work describes the development of a model in Artificial Neural Networks (ANN) with the objective of simulating the capacity of a river to recover the amount of dissolved oxygen (OD), after receiving a polluting load. The experiment was carried out in a river located in the municipality of Medianeira in the State of Paraná, which receives the treated effluent from a slaughterhouse agroindustry. The model was developed based on data on the water quality of the river and the effluent that is incorporated into the watercourse over the studied interval. For training and validation of the model, data from previous studies, collected in 2014, were used. The input variables that were used in the network are the parameters of water and effluent quality and the output was the OD. To this end, five RNA models were developed, which underwent modifications in the input variables, but maintained the same output and the results simulated by the RNA were compared with the results collected in loco. In addition, the results obtained were compared with the results of a study carried out in 2014, with the objective of verifying the evolution of the technologies used in ANN. Among the main results of the present study, the processing speed stands out in relation to the previous study. Among the developed models, the model named RNA3 was the one that presented results more similar to the results collected in the field, demonstrating greater capacity for simulating the self-purification of the river.

Keywords: water quality; DBO; OD; artificial neural networks.

LISTA DE FIGURAS

Figura 1 - Estágios ecológicos divididos em zonas identificáveis nos rios.	20
Figura 2 - descrição do k-ésimo neurônio	22
Figura 3 - Exemplo de uma rede neural artificial	23
Figura 4 - Conexão entre os neurônios “j” e “k”	26
Figura 5 - Localização do Rio alegria e seus afluentes.	32
Figura 6 - Malha municipal simplificada	33
Figura 7 - Pontos de amostragem e coleta de água no Rio Alegria	34
Figura 8 - Classificação metodológica da pesquisa.	34
Figura 9 - Fluxograma metodológico do estudo	36
Figura 10 - Modelo RNA1	37
Figura 11 - Erro médio RNA1 (MAE) e (MSE).	38
Figura 12 - Valores de OD reais x OD Previstos RNA1.	39
Figura 13 - Treinamento do modelo com <i>EarlyStopping</i>	39
Figura 14 - RNA1 MAE e MSE com <i>EarlyStopping</i>	40
Figura 15 - Comparação OD real x Calculado RNA1	40
Figura 16 - Gráficos gerados na validação cruzada do primeiro modelo RNA1 Oxigênio dissolvido real e calcula	41
Figura 17 - Modelo RNA2	42
Figura 18 - Erro médio RNA2 (MAE) e (MSE).	42
Figura 19 - Valores de OD reais x OD Previstos RNA2.	43
Figura 20 - RNA2 MAE e MSE com <i>EarlyStopping</i>	43
Figura 21 - Comparação OD real x Calculado RNA2	44
Figura 22 - Comparação entre os dados conhecidos e os dados simulados pela rede RNA2	44
Figura 23 - Modelo RNA3	45
Figura 24 - Erro médio RNA3 (MAE) e (MSE).	45
Figura 25 - Valores de OD reais x OD Previstos RNA3.	46
Figura 26 - RNA3 MAE e MSE com <i>EarlyStopping</i>	46
Figura 27 - Comparação OD real x Calculado RNA3	47
Figura 28 - Comparação entre os dados conhecidos e os dados simulados pela rede RNA3	47
Figura 29 - Modelo RNA4	48
Figura 30 - Erro médio RNA4 (MAE) e (MSE).	48
Figura 31 - Valores de OD reais x OD Previstos RNA4.	49
Figura 32 - RNA4 MAE e MSE com <i>EarlyStopping</i>	49
Figura 33 - Comparação OD real x Calculado RNA4	50
Figura 34 - Comparação entre os dados conhecidos e os dados simulados pela rede RNA4	50
Figura 35 - Modelo RNA5	51
Figura 36 - Erro médio RNA5 (MAE) e (MSE).	51
Figura 37 - Valores de OD reais x OD Previstos RNA5.	52

Figura 38 - RNA5 MAE e MSE com <i>EarlyStopping</i>	52
Figura 39 - Comparação OD real x calculado RNA5.....	53
Figura 40 - Comparação entre os dados conhecidos e os dados simulados pela rede RNA5.....	53

LISTA DE TABELAS

Tabela 1 – Comparativo de resultados entres as RNA's	54
--	----

LISTA DE ABREVIATURAS E SIGLAS

API	<i>Application Programming Interface</i>
CE	Condutividade elétrica
DBO	Demanda bioquímica de oxigênio
DQO	Demanda química de Oxigênio
IA	Inteligência Artificial
Na	Nitrogênio amoniacal
NI	Nitrito
OD	Oxigênio Dissolvido
PH	potencial hidrogeniônico
RNA	Redes Neurais Artificiais
T	Temperatura

SUMÁRIO

1 INTRODUÇÃO	10
1.1 OBJETIVO GERAL	13
1.2 OBJETIVOS ESPECÍFICOS	13
2 FUNDAMENTAÇÃO TEÓRICA	14
2.1 QUALIDADE DA ÁGUA.....	14
2.1.1 Demanda Bioquímica De Oxigênio	14
2.1.2 Potencial Hidrogeniônico (pH)	15
2.1.3 Temperatura	16
2.1.4 Condutividade elétrica	16
2.1.5 Oxigênio dissolvido	17
2.2 IMPORTÂNCIA DA AUTODEPURAÇÃO	18
2.3 REDES NEURAIS ARTIFICIAIS	21
2.4 GERENCIADOR DE PACOTES ANACONDA	28
2.5 IDE SPYDER.....	29
2.6 FRAMEWORK TENSORFLOW	29
2.7 API KERAS	30
2.8 LINGUAGEM DE PROGRAMAÇÃO PYTHON	31
3 PROCEDIMENTOS METODOLÓGICOS DA PESQUISA	32
3.1 ÁREA DE ESTUDO.....	32
3.2 DADOS DE COLETA	33
3.3 TIPO DE PESQUISA OU TÉCNICAS DE PESQUISA	34
3.4 FERRAMENTAS UTILIZADAS.....	35
4 RESULTADOS	37
5 CONCLUSÃO	55
REFERÊNCIAS	56
APÊNDICE A- Banco de dados de treinamento	62
APÊNDICE B- Agoritimo RNA1	66

1 INTRODUÇÃO

Na relação direta com o entorno, o lançamento de carga orgânica solúvel biodegradável por indústrias no receptor, exige avaliação previa das consequências sobre a qualidade do ambiente aquático, com preferencial acompanhamento experimental subsequente. Neste sentido, estudos preliminares para entendimento dos fenômenos de dispersão e autodepuração das espécies químicas de interesse permitem inferências sobre a distribuição de concentrações e possíveis consequências sobre a vida aquática. Para a solução analítica e obtenção de respostas, a adoção de hipóteses simplificadoras severas é necessária, como a condição unidimensional do fluxo, a idealização de geometrias e a validade de constantes experimentais.

Neste viés, modelos de dispersão para avaliação do comportamento da dispersão de poluentes e do consumo de oxigênio na depuração da carga orgânica fornecem indicativos dos efeitos de lançamento. Entretanto, dados os significativos desvios de idealidade próprios do ambiente natural, demandam posterior acompanhamento experimental.

Estudos acadêmicos sobre seus níveis de comprometimento são extremamente importantes, com monitoramento de parâmetros relevantes para enquadramento qualitativo como DBO e oxigênio dissolvido. Disto, dados experimentais tornam-se passíveis de apropriação para refinamento dos modelos de simulação. Visto que o principal indicador da qualidade do ambiente aquático, a concentração de oxigênio dissolvido no receptor é reduzida por conta da oxidação biológica de compostos orgânicos biodegradáveis.

Conhecer as concentrações médias de montante e de jusante, acompanhadas das vazões e condições de lançamento, permitem o estabelecimento do balanço de massa das condições de diluição, e tomadas de decisão, pois convergem com os referenciais teóricos sobre o tema e permitem identificar a influência da elevação da DQO (nos níveis existentes) sobre a concentração de oxigênio dissolvido e condições de depuração.

A concentração de oxigênio dissolvido (DO) reflete o equilíbrio entre a produção de oxigênio (por exemplo, fotossíntese) e o consumo de oxigênio (por exemplo, respiração aeróbica, nitrificação e produtos químicos oxidação) em ecossistemas

aquáticos. Dependem de muitos fatores como temperatura, salinidade, depleção de oxigênio, oxigênio fontes e outros. O nível de OD é o critério de saúde que é frequentemente usado para controle de qualidade em diferentes sistemas aquáticos, como reservatórios e zonas húmidas (OLYAIE; ABYANEH; MEHR, 2017).

Devido à característica dinâmica da concentração de OD, especialmente em rios e zonas úmidas, é altamente recomendável gerar modelos de OD periodicamente para os ecossistemas aquáticos, para que medidas de controle de qualidade pode ser otimizado ao longo de um horizonte de tempo. Para este fim, implementação de diferentes técnicas de inteligência artificial (IA) foram sugeridos na literatura relevante.

Em relação à previsão dos parâmetros de qualidade da água (WQPs), redes neurais (RNAs) foram bem-sucedidas aplicadas para a estimativa de temperatura, cloreto, fluoreto, condutividade elétrica, alcalinidade, dureza total, salinidade, sólidos totais dissolvidos, razão de adsorção de sódio, nitrogênio amoniacal, bicarbonato, produto químico e demanda biológica de oxigênio (DQO e DBO), oxigênio dissolvido (DO), porcentagem de OD (MITROVIĆ et al., 2019).

Os modelos de RNA têm se mostrado importante na de previsão para a relação entre os parâmetros de precipitação e de escoamento. Os resultados apoiarão a tomada de decisões na área de planejamento e gerenciamento de recursos hídricos. Além disso, ajudam os planejadores e gestores urbanos a tomar as medidas necessárias para enfrentar as más produções. Assim, eles ajudam a evitar perdas em propriedades públicas e privadas e riscos à saúde e ao meio ambiente que provavelmente ocorrerão devido a inundações. (AICHOURI et al., 2015).

Devido à sua arquitetura de processamento massivamente paralela, a RNA é capaz de lidar eficientemente com cálculos complexos, tornando-a hoje a técnica mais preferida para alta velocidade processamento de grandes dados. Além disso, existem muitas características vantajosas da abordagem da RNA para o problema.

A aplicação de uma rede neural não requer conhecimento a priori do processo subjacente, um pode não reconhecer todas as complexas relações existentes entre os vários aspectos do processo sob investigação, uma abordagem de otimização padrão ou modelo estatístico fornece uma solução somente quando é permitido executar até a conclusão considerando que uma rede neural converge sempre para uma condição ótima da solução e, nem restrições ou uma estrutura de solução a principio são necessariamente assumidas ou estritamente aplicadas no desenvolvimento da RNA (essas características tornam as RNAs ferramentas muito

adequadas para lidar com vários problemas de modelagem hidrológica) (SALARI; SHAHID; AFZALI; EHTESHAMI; CONTI; DERAKHSAN; SHEIBANI, 2018).

No entanto, o número de usos para RNAs está aumentando rapidamente e, nos últimos anos, eles foram usados com sucesso para a previsão de economia, recursos hídricos, qualidade da água e séries temporais hidrológicas (SARKAR; PANDEY, 2015).

Desde os anos 90, com base no entendimento do cérebro e do sistema nervoso, redes neurais artificiais (RNAs) têm sido gradualmente usadas em previsões hidrológicas. Uma RNA aprende a resolver um problema desenvolvendo uma memória capaz de correlacionar um grande número de padrões de entrada com um conjunto resultante de rendimentos (OLYAIE; ABYANEH; MEHR, 2017).

1.1 OBJETIVO GERAL

Desenvolver um modelo em redes neurais artificiais para simular o Oxigênio Dissolvido em rios.

1.2 OBJETIVOS ESPECÍFICOS

- a. Analisar o banco de dados do estudo de (SCHÜTZ,2014) para treinamento e validação.
- b. Treinamento de cinco modelos distintos de RNA com variação nas estruturas das redes, conforme os modelos proposto em Schütz 2014;
- c. Validação dos resultados dos cinco modelos desenvolvidos.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 QUALIDADE DA ÁGUA

A adulteração da qualidade natural da água pela atuação do homem se delibera como poluição, e faz com que seja parcial ou totalmente imprópria para o uso a que se propõe. Entende-se por qualidade natural da água o conjunto de particularidades físicas, químicas e bacteriológicas que apresenta a água em seu estado natural nos rios, lagos, mananciais, no subsolo ou no mar. O estabelecimento das condições e padrões de lançamento de efluentes em corpos de água receptores é disposto pela Resolução 430/2011 do Conselho Nacional do Meio Ambiente (CONAMA), que complementa e altera a Resolução 357/2005 (SCHÜTZ, 2014).

Além da atividade humana, em cada etapa do ciclo hidrológico a água entra em contato e, naturalmente, incorpora em sua composição, elementos existentes na atmosfera, solos, vegetação e rochas.

Desta forma, a água que escoar na superfície e no subsolo passa a ter impurezas orgânicas, inorgânicas como cálcio, magnésio, sódio, bicarbonatos, cloretos, sulfatos e nitratos e também traços de alguns metais como chumbo, cobre e manganês. Durante as décadas passadas houve um agravamento dos problemas qualitativos dos recursos hídricos pela contaminação. A qualidade da água pode ser utilizada como diagnóstico do estado de preservação do ambiente, além de influir diretamente na saúde humana. As condições climáticas das diferentes regiões brasileiras exigem um tratamento diferenciado em relação à fluoretação da água de abastecimento público.

2.1.1 Demanda Bioquímica De Oxigênio

A Demanda Bioquímica de Oxigênio (DBO) é a quantidade de oxigênio molecular necessário à estabilização da matéria orgânica decomposta aerobiamente por via

biológica. É utilizada para exprimir o valor da poluição produzida por matéria orgânica oxidável biologicamente, que corresponde à quantidade de oxigênio que é consumida pelos microrganismos do esgoto ou águas poluídas, na oxidação biológica, quando mantida a uma dada temperatura por um espaço de tempo convencionado (SCHUTZ, 2014).

Segundo Dutra (2014), os processos físicos, químicos e biológicos interferem na dinâmica de variação das concentrações de Oxigênio Dissolvido (OD) no corpo receptor, bem como os mecanismos bioquímicos relacionados ao consumo de OD, responsável pela variação da Demanda Bioquímica de Oxigênio. Ambos os parâmetros são universalmente utilizados como indicadores da presença da matéria orgânica aportada ao corpo receptor e estão relacionados ao seu grau de poluição.

A DBO é a quantidade de oxigênio consumido durante um determinado período de tempo, numa temperatura de incubação específica. Neste processo, no metabolismo dos microrganismos heterotróficos, os compostos orgânicos biodegradáveis são transformados em produtos finais estáveis ou mineralizados e há um consumo de oxigênio da água e liberação da energia contida nas ligações químicas das moléculas decompostas.

2.1.2 Potencial Hidrogeniônico (pH)

O termo pH (Potencial Hidrogeniônico) é usado para expressar a intensidade da condição ácida ou básica de uma solução e a concentração do íon hidrogênio. O pH é muito influenciado pela quantidade de matéria morta a ser decomposta, sendo que quanto maior a quantidade de matéria orgânica disponível, menor o pH, pois para haver decomposição de matérias, muitos ácidos são produzidos (SCHUTZ, 2014).

A escala de pH é constituída de uma série de números naturais que varia entre 0 e 14, e denota vários graus de acidez e basicidade de uma solução ou composto, ou seja, é o modo de expressar a concentração de íons de hidrogênio em uma solução. Valores inferiores a 7 são considerados valores para soluções ácidas, enquanto valores acima de 7, até o máximo de 14, indicam aumento da basicidade. É um parâmetro de caráter operacional que deve ser acompanhado para aperfeiçoar os

processos de tratamento da água e preservar as tubulações contra corrosões ou entupimentos (DUTRA, 2014).

O valor do pH influencia na distribuição das formas livre e ionizada de diversos compostos químicos, além de contribuir para um maior ou menor grau de solubilidade das substâncias e de definir o potencial de toxicidade de vários elementos (ESTEVES, 1998).

2.1.3 Temperatura

É um parâmetro importante, pois influi em algumas propriedades da água, tais como densidade, oxigênio dissolvido, viscosidade, com reflexos sobre a vida aquática. Em ambientes lênticos, quando as diferenças de temperatura geram camadas d'água com diferentes densidades, que em si já formam barreira física impedindo que se misturem, e adicionalmente, se a energia do vento não for suficiente para misturá-las, o calor não se distribui uniformemente criando a condição de estabilidade térmica (DUTRA, 2014).

A temperatura desempenha um papel crucial no meio aquático, condicionando as influências de uma série de variáveis físico-químicas. Em geral, à medida que a temperatura aumenta, de 0 a 30°C, viscosidade, tensão superficial, compressibilidade, calor específico, constante de ionização e calor latente de vaporização diminuem, enquanto a condutividade térmica e a pressão de vapor aumentam (SCHUTZ, 2014).

2.1.4 Condutividade elétrica

O parâmetro condutividade elétrica (CE) é uma medida resultante da aplicação de uma dada força elétrica, que é diretamente proporcional à quantidade de sais presentes em uma solução, conseqüentemente, esse parâmetro expressa a concentração total de sais para classificação de solos e das águas destinadas à

irrigação. Quanto maior for à quantidade de íons dissolvidos, maior foi a condutividade elétrica da água.(SCHUTZ, 2014).

Este parâmetro não determina, especificamente quais íons estão presentes em determinada amostra, mas pode contribuir para possíveis reconhecimentos de impactos ambientais. A condutividade elétrica da água pode variar de acordo com a temperatura e a concentração total de substâncias ionizadas dissolvidas.

Em águas cujos valores de pH se localizam nas faixas extremas ($\text{pH} > 9$ e $\text{pH} < 5$), os valores de condutividade são devidos apenas às altas concentrações de poucos íons em solução, dentre os quais os mais frequentes são H^+ e o OH^- -(DUTRA, 2014).

2.1.5 Oxigênio dissolvido

A oferta de oxigênio dissolvido (OD) em águas superficiais é vital para toda a vida aquática desse sistema. Os problemas associados à baixas concentrações de OD em rios são conhecidos à muito tempo, de forma que o ecossistema fica desequilibrado, ocasionando a mortalidade de peixes e presença de odores. Isso causa grande incômodo na população que habita nas proximidades. Um exemplo é o rio Pinheiros na cidade de São Paulo, o qual tem aparelhos de transporte nas suas margens (trem e avenidas) e apresenta forte cheiro em grande parte do tempo (TERCINI, 2014).

A determinação do OD é de fundamental importância para avaliar as condições naturais da água e detectar impactos ambientais como eutrofização e poluição orgânica. Geralmente o oxigênio dissolvido se reduz ou desaparece, quando a água recebe grandes quantidades de substâncias orgânicas biodegradáveis encontradas, por exemplo, no esgoto doméstico, em certos resíduos industriais, e outros, o que auxilia no monitoramento da qualidade de efluentes industriais (SCHUTZ, 2014).

2.2 IMPORTÂNCIA DA AUTODEPURAÇÃO

O processo de autodepuração, segundo Tercini (2014), é responsável pela recuperação dos níveis de oxigênio dissolvido nas águas correntes e está basicamente correlacionado com a decomposição biológica que ocorre naturalmente nos cursos d'água. As características fundamentais para a vida existente em um rio são associadas primordialmente pela quantidade de oxigênio dissolvido no seu meio e pela sua capacidade em reduzir a poluição orgânica através de processos naturais, físicos e bioquímicos.

O aporte de matéria orgânica consome oxigênio da água pelos processos de metabolismo microbiano. A recuperação natural dos níveis de oxigênio da água (autodepuração) ocorre em função do gradiente de concentração de gases presentes entre a interface ar-água (K_2). Esse processo é acelerado quando ocorre intensa turbulência do corpo receptor (aumento de velocidades), que permite maior superfície de contato entre a massa líquida e o ar atmosférico que, por seu turno, permite maior troca gasosa entre a atmosfera e a água (DUTRA, 2014).

Calmon (2015) realizou uma análise conjunta de curvas de permanência de qualidade e capacidade de autodepuração de rios. Para composição dos diferentes cenários de simulação da qualidade da água, foram estabelecidos no referido trabalho três panoramas de tratamento de esgotos associados com distintas eficiências de remoção de demanda bioquímica de oxigênio (DBO), duas condições de abatimento de cargas orgânicas e três horizontes de análise: 2014, 2020 e 2030. Foram produzidas curvas de permanência de qualidade para o parâmetro DBO₅, 20, associadas aos padrões ambientais estabelecidos pela Resolução CONAMA Nº 357/2005 para rios Classes 1, 2 e 3. Os resultados obtidos mostraram a relevância da incorporação da modelagem da autodepuração na avaliação da qualidade dos corpos d'água superficiais para a implementação do enquadramento e, adicionalmente, indicam que a visão não determinística, oferecida pelas curvas de permanência, facilitaria o processo de tomada de decisão acerca do estabelecimento de metas progressivas para a efetivação do enquadramento.

Salla (2013) avaliou a capacidade de autodepuração do Rio Jordão, considerando as contribuições reais do córrego Brejo Alegre em período de estiagem

e os dois cenários que levam em conta as cargas poluidoras estimadas da estação de tratamento de esgoto e a baixa capacidade de diluição do rio para a vazão crítica Q7, 10, por meio do modelo QUAL-UFMG. A simulação na estiagem apresentou aceitável calibração, tendo em vista a otimização dos coeficientes, evidenciando a prevalência da desoxigenação por demanda carbonácea sobre a nitrificação e a importância da reaeração natural no processo de autodepuração.

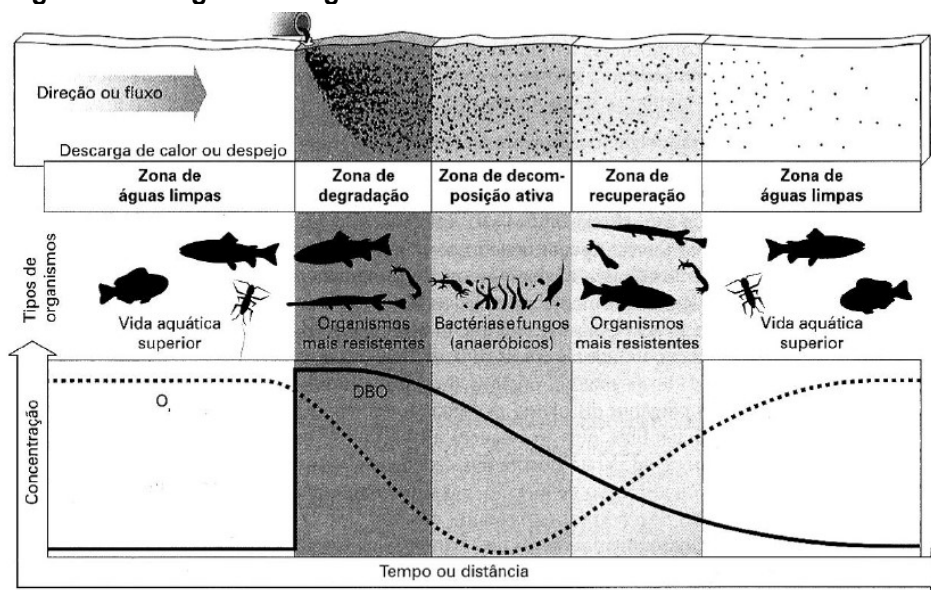
Fantin (2017), estabeleceu a metodologia para a pré-seleção de alternativas de tratamento de esgoto no âmbito de bacias hidrográficas. A proposta metodológica envolveu o uso combinado de modelo de qualidade de água, técnica de otimização e um conjunto de critérios de natureza técnica e econômica associados aos diferentes sistemas de tratamento de esgotos. Os resultados indicaram que o modelo de otimização que buscou a minimização das eficiências de tratamento no âmbito da bacia, permitindo uso da capacidade e de autodepuração dos cursos d'água, selecionou sistemas de tratamento que variaram da combinação de reatores UASB e lagoas de polimento, à sistemas de infiltração lenta, alternativas de tratamento que apresentaram menores estimativas de custos presentes líquidos. A incorporação da equidade entre os sistemas de tratamento aumentou significativamente os custos presentes líquidos totais associados ao tratamento de esgoto no âmbito da bacia hidrográfica.

A autodepuração pode ser definida como a capacidade de um rio ou curso de água de recuperar o equilíbrio, por meio de formas naturais, após a inserção de cargas orgânicas ou poluidoras. Isto não quer dizer, necessariamente, que as águas retornam as condições iniciais, mas que, o ecossistema busca um equilíbrio, sem apresentar problemas ambientais.

“Há uma sequencia sistemática de substituição de uma comunidade por outra, até que uma comunidade estável se estabeleça em equilíbrio com as condições locais” (VON SPERLING, 1996, v. 1, p.94).

Ao longo do tempo e considerado o perfil longitudinal dos rios, existem etapas de sucessões ecológicas associadas a zonas fisicamente identificáveis nos cursos de águas, chamadas zonas de autodepuração, conforme Figura 1.

Figura 1 - Estágios ecológicos divididos em zonas identificáveis nos rios.



Fonte: (BRAGA, 2005).

Segundo Braga *et al* (2005), são quatro zonas sendo:

- Zona de degradação: tem início logo após o lançamento dos efluentes no curso de água, apresenta alta concentração de matéria orgânica em estágio complexo, mas com potencial de decomposição;
- Zona de decomposição ativa: o ecossistema tenta se organizar após a perturbação, com microrganismos desempenhando ativamente a decomposição da matéria orgânica. A qualidade das águas apresenta-se em seu pior estágio;
- Zona de recuperação: como o próprio nome diz, inicia-se a etapa de recuperação. A água fica mais clara e a maior parte da matéria orgânica encontra-se estabilizada;
- Zona de águas limpas: retornam as condições normais anteriores a poluição com relação ao oxigênio dissolvido, à matéria orgânica, bactérias e patogênicos.

2.3 REDES NEURAIIS ARTIFICIAIS

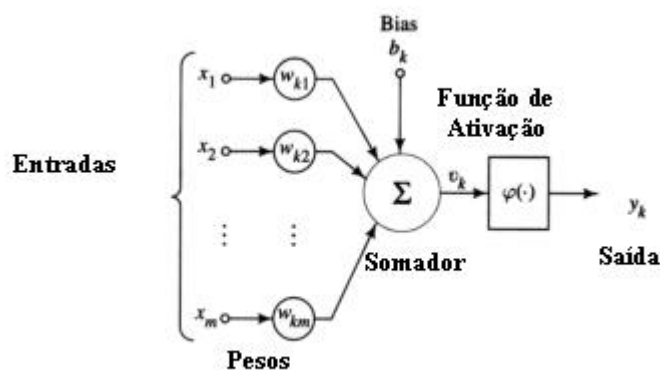
A expressão “rede neural” é devido à tentativa dos modelos imitarem a capacidade que o cérebro humano possui de reconhecer, associar e generalizar padrões. Corresponde a uma importante técnica estatística não linear capaz de resolver uma gama de problemas complexos. Isso torna o método extremamente útil quando não é possível definir um modelo explícito ou uma lista de regras. Em geral, isso acontece em situações em que o ambiente dos dados muda muito. As principais áreas de atuação são para a classificação de padrões e regressão (JAIN,1996).

Para Haykin (2008) o trabalho em redes neurais artificiais, tem sido motivado desde o começo pelo reconhecimento de que o cérebro humano processa informações de uma forma inteiramente diferente do computador convencional. O cérebro é um *computador* (sistema de processamento de informação) altamente complexo, não-linear, e paralelo. Ele tem a capacidade de organizar seus constituintes estruturais, conhecidos por neurônios, de forma a realizar certos processamentos (p.ex., reconhecimentos de padrões, percepção e controle motor) muito mais rapidamente que qualquer computador hoje existente.

Os neurônios são elementos processadores interligados, trabalhando em paralelo para desempenhar uma determinada tarefa. Em geral, isso acontece quando o ambiente gerador dos dados muda constantemente. As principais áreas de atuação são para classificação de padrões e previsão.

O ponto de partida é definir o que são e como se constituem as suas unidades básicas. Nesta perspectiva, a figura 2 mostra a descrição funcional do k-ésimo neurônio de uma rede, ou seja, descreve o que se encontra no interior do neurônio.

Figura 2 - descrição do k-ésimo neurônio



Fonte: (VELLASCO, 2007).

As entradas estão representadas pelos x 's. Esses x 's não são os padrões da camada de entrada, mas a saída do neurônio da camada anterior.

Bias (Tendência) é um erro sistemático diferentemente de erro aleatório. Um ou mais componentes do erro sistemático podem contribuir para a tendência. Uma grande diferença sistemática em relação ao valor de referência aceito reflete-se em um grande valor de tendência.

Os w 's representam os pesos ou os parâmetros da rede. Estes pesos representam a "memória" da rede. Isto é, a experiência ganha como resultado das n -apresentações dos padrões. São os pesos que combinam a não-linearidade para que a mesma fique distribuída pela rede.

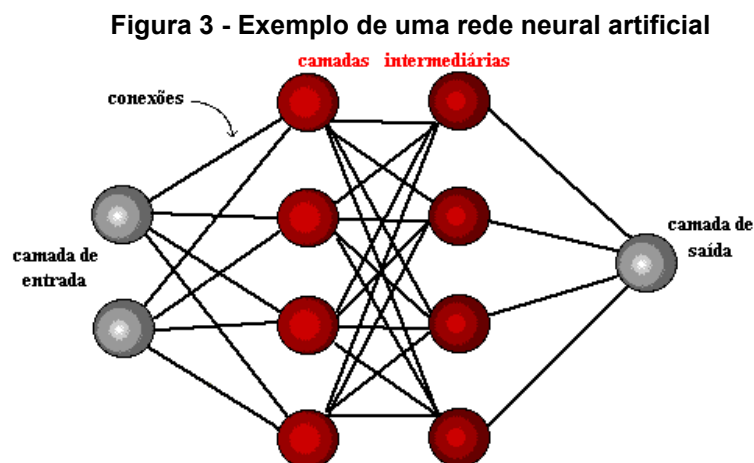
U_k (net_k) representa a combinação linear dos pesos. Corresponde a soma ponderada da entrada pelos pesos. Y_k é a saída do k-ésimo neurônio que depende do nível de ativação aplicado ao neurônio pela função de ativação.

Ressalta-se que a função de ativação refere-se a parte não-linear de cada neurônio, sendo o único lugar que a não-linearidade se encontra.

θ_k é conhecido como termo polarizador. Indica o ponto em que a função se encontra em cima do eixo e, portanto, define o domínio dos valores de saída. Na modelagem recorre-se ao artifício de tratar este termo como mais um peso de modo que, durante o processo de otimização dos pesos, a ser realizado pelo algoritmo implementado, a atualização aconteça para todos os parâmetros, incluindo para o polarizador.

Em uma rede neural os parâmetros a serem estimados são os pesos e o polarizador. Como em cada neurônio chega a soma ponderada de todas as entradas, então o polarizador aparecerá associado a uma entrada fixa +1 ou -1.

Através da rede passam dois tipos de sinais, sendo o primeiro deles chamado de sinal função. Este sinal é recebido pelos nódulos da camada de entrada na forma de estímulos gerados pelo ambiente externo e propaga-se ao longo da rede, sendo processado em cada neurônio intermediário na forma de uma função das entradas, ponderadas pelos pesos à elas associados, até que emergindo dos nódulos da camada de saída, produz a resposta da rede. Já o segundo sinal, é constituído de uma função erro, que se propaga no sentido oposto à rede, sendo útil para o algoritmo de aprendizagem Backpropagation (Retropropagação). (EYNG, 2008).



Fonte BARRETO (2005)

A retropropagação é interna a rede, acontece em pelo menos uma camada de neurônios e não está envolvida com a entrada ou saída. A camada que é submetida a retropropagação e suas conexões quando, aprendem a efetuar uma função, agem como se houvesse uma representação interna da solução do problema.

Matematicamente, o erro na iteração n , correspondente à análise do n -ésimo par de exemplos (dados de entrada – saída) apresentados à rede, é a diferença entre a resposta atual, $y_j(n)$, e a desejada, $d_j(n)$, conforme equação 1.1.

$$e_j(n) = d_j(n) - y_j(n) \quad (1.1)$$

O erro calculado em relação à todos os M nódulos de saída pode ser obtido pela equação 1.2.

$$E_n = \frac{1}{2} \sum_{j=1}^M e_j^2(n) \quad (1.2)$$

A função objetivo, que se deseja minimizar é dada pela Equação 1.3.

$$\bar{E} = \frac{1}{N} \sum_{n=1}^N E(n) \quad (1.3)$$

Em que N representa o número total de pares apresentados.

O neurônio j seja alimentado pelos sinais função $y_i(n)$ emitidos pelos nódulos i da camada anterior, sendo cada sinal associado à um peso sináptico $w_{ij}(n)$, o nível de atividade do nódulo j pode ser expresso pela Equação 1.4.

$$v_j(n) = \sum_{i=0}^I w_{ij}(n) y_i(n) \quad (1.4)$$

Em que I é o número total de nódulos que enviam sinais função ao nódulo j, e $w_{0j}(n)$ corresponde ao limiar de j.

Desta forma, a resposta produzida pela rede foi apresentada na Equação 1.5.

$$y_j(n) = \varphi(v_j(n)) \quad (1.5)$$

Em que φ é uma função contínua e diferenciável. O algoritmo de retropropagação baseia-se no ajuste dos pesos sinápticos proporcionalmente à diminuição do erro, $E(n)$. Assim, através da regra da cadeia pode-se escrever a Equação 1.6.

$$\frac{\partial E(n)}{\partial w_{ij}(n)} = \left(\frac{\partial E(n)}{\partial e_j(n)} \right) \left(\frac{\partial e_j(n)}{\partial y_j(n)} \right) \left(\frac{\partial y_j(n)}{\partial v_j(n)} \right) \left(\frac{\partial v_j(n)}{\partial w_{ij}(n)} \right) \quad (1.6)$$

Diferenciando a Equação 1.2 em relação à $e_j(n)$, têm-se:

$$\frac{\partial E(n)}{\partial e_j(n)} = e_j(n) \quad (1.7)$$

Diferenciando a Equação 1.1 em relação à $y(n)$, têm-se:

$$\left(\frac{\partial v_j(n)}{\partial w_{ij}(n)} \right) = y_i(n) \quad (1.8)$$

O terceiro termo da Equação 1.7 é obtido por meio da diferenciação da Equação 1.5 em relação à $v_j(n)$.

$$\left(\frac{\partial v_j(n)}{\partial w_{ij}(n)} \right) = y_i(n) \quad (1.9)$$

Finalmente, diferenciando a Equação 1.4 em relação à $w_{ij}(n)$, têm-se:

$$\left(\frac{\partial E_j(n)}{\partial w_{ij}(n)} \right) = -e_j(n)\varphi'(v_j(n))y_i(n) \quad (1.10)$$

A correção dos pesos sinápticos, $\Delta w_{ij}(n)$ é obtida por meio da regra delta.

$$\Delta w_{ij}(n) = -\eta \left(\frac{\partial E_j(n)}{\partial w_{ij}(n)} \right) \quad (1.11)$$

Em que η é o parâmetro da taxa de aprendizagem, e o uso do sinal negativo deve-se ao fato do aprendizado ser efetuado no sentido do decréscimo do gradiente do erro. Substituindo-se a Equação 1.11 na Equação 1.10, têm-se:

$$\Delta w_{ij}(n) = \eta \delta_i(n) y_j(n) \quad (1.12)$$

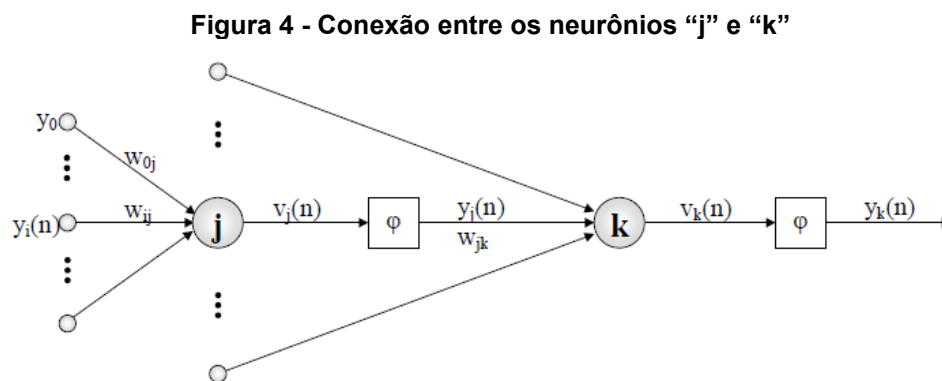
Em que:

$$\delta_j(n) = -\frac{\partial E_j(n)}{\partial v_j(n)} = e_j(n)\varphi_j'(v_j(n)) \quad (1.13)$$

Observando-se a Equação 1.13 pode-se concluir que a dificuldade de se definir o ajuste dos pesos está concentrada no cálculo do erro, $e_j(n)$. Se j é um nódulo de saída, então o cálculo do erro é extremamente simples, pois a resposta desejada é conhecida, o que não acontece quando o neurônio j está localizado em uma camada oculta. Neste caso, o erro deve ser determinado a partir de sinais provenientes dos neurônios da camada seguinte que estão conectados ao nódulo j . Aplicando a regra da cadeia na Equação 1.13, têm-se:

$$\delta_j(n) = - \left(\frac{\partial E_j(n)}{\partial v_j(n)} \right) \left(\frac{\partial y_j(n)}{\partial v_j(n)} \right) = - \left(\frac{\partial E(n)}{\partial y_j(n)} \right) \varphi_j'(v_j(n)) \quad (1.14)$$

Considerando um neurônio k , integrante da camada de saída, conforme demonstrado na Figura, pode-se proceder ao cálculo do erro segundo a Equação 1.15.



Fonte: EYNG (2008).

$$E_n = \frac{1}{2} \sum_k \sum e_k^2(n) \quad (1.15)$$

Diferenciando a Equação 1.15 em relação à $y_j(n)$, têm-se:

$$\frac{\partial E(n)}{\partial y_j(n)} = \sum_k e_k \left(\frac{\partial e_k(n)}{\partial y_j(n)} \right) \quad (1.16)$$

Aplicando a regra da cadeia à Equação 1.16, têm-se:

$$\frac{\partial E(n)}{\partial y_j(n)} = \sum_k e_k(n) \left(\frac{\partial e_k(n)}{\partial v_j(n)} \right) \left(\frac{\partial v_j(n)}{\partial y_j(n)} \right) \quad (1.17)$$

Como o erro na última camada pode ser calculado com base na resposta desejada, para o nóculo k , têm-se:

$$e_k(n) = d_k(n) - y_k(n) = d_k(n) - \varphi_k(v_k(n)) \quad (1.18)$$

Diferenciando a Equação 1.18 em relação à $v_k(n)$, têm-se:

$$\left(\frac{\partial e_k(n)}{\partial v_j(n)} \right) = -\varphi_k'(v_k(n)) \quad (1.19)$$

O nível de atividade para o nóculo k é dado pela Equação 1.20.

$$v_k(n) = \sum_{j=0}^m w_{jk}(n) y_j(n) \quad (1.20)$$

Diferenciando a Equação 1.20 em relação à $y_j(n)$ têm-se:

$$\left(\frac{\partial v_k(n)}{\partial y_j(n)} \right) = w_{jk}(n) \quad (1.21)$$

Substituindo-se as Equações 1.21 e 1.19 na Equação 1.16, obtém-se a relação desejada.

$$\frac{\partial E(n)}{\partial v_j(n)} = -\sum_k e_k(n) \varphi_k'(v_k(n)) w_{jk}(n) = \sum_k \delta_k(n) w_{jk}(n) \quad (1.22)$$

Finalmente substituindo-se a Equação 1.22 na Equação 1.14, obtém-se a fórmula da retropropagação para $\delta_j(n)$, no caso em que j é um nóculo oculto.

$$\delta_j(n) = \varphi_j'(v_j(n)) \sum_k \delta_k(n) w_{jk}(n) \quad (1.23)$$

Assim, é possível determinar o ajuste dos pesos sinápticos vinculados às conexões com o neurônio j , seja ele integrante de uma camada intermediária ou da camada de saída. A taxa de aprendizagem (η), presente na Equação 1.11, constitui um importante parâmetro, pois estabelece uma relação entre a velocidade de convergência e a probabilidade de instabilidade no processo de aprendizagem. Caso η assumira valores muito pequenos, as mudanças nos pesos sinápticos também serão muito suaves, acarretando assim, um aumento do número de iterações requeridas pelo treinamento. Entretanto, se η for muito grande, as mudanças serão bruscas, provocando instabilidades no processo de aprendizagem.

2.4 GERENCIADOR DE PACOTES ANACONDA

De acordo com o site do distribuidor Anaconda Enterprise (2019), “[...] a Anaconda é a plataforma de dados científicos mais popular no mundo e a base do moderno aprendizado de máquina”, sendo um gerenciador de pacotes para ciência de dados e IA. Com ela é possível criar ambientes com versões específicas de cada pacote que irá utilizar em seu algoritmo, garantindo a compatibilidade entre eles.

É possível realizar tudo a partir da plataforma Anaconda, sem envolver DevOps (Desenvolvimento e Operações). Dessa forma, possibilita aos cientistas de dados a desenvolver e implantar os modelos de IA rapidamente na produção, pois a plataforma fornece as ferramentas necessárias para a coleta de dados diretamente de arquivos, banco de dados ou lago de dados, bem como o gerenciamento de ambientes com o

Conda (terminal de comando da Anaconda), o compartilhamento e colaboração de projetos, além da implantação de projetos em produção com apenas um único clique de um botão. (ANACONDA ENTERPRISE, 2019)

2.5 IDE SPYDER

O Spyder é um Ambiente de Desenvolvimento Integrado (IDE) escrito em Python e para Python, sendo um poderoso ambiente científico para analistas, engenheiros e cientistas de dados. Em uma única tela, possibilita a utilização de recursos avançados de edição, análise e depuração de dados, acelerando o desenvolvimento e a implementação de algoritmos funcionais. (SPYDER IDE, 2019)

O Spyder IDE está disponível para download através da plataforma Anaconda, tornando-o versátil quanto a sua instalação e funcionamento, uma vez que ele pode trabalhar em diversos ambientes distintos gerenciados através da plataforma.

2.6 FRAMEWORK TENSORFLOW

De acordo com Roberts e Leichenauer (2019), o Google, a fim de utilizar-se de redes tensoras para resolver os problemas científicos mais difíceis do mundo real, que acabam envolvendo lidar com a complexidade de sistemas quânticos, que em alguns casos tem estados exponenciais, o que inviabiliza a computação de força bruta, mas que é possível de ser resolvido a partir das redes tensoras. Essas redes tensoras permitem o enfoque nos estados quânticos de maior relevância para os problemas do mundo real (estados de baixa energia), enquanto ignoram os estados menos relevantes.

Segundo o distribuidor TensorFlow (2019), ele é “[...] uma plataforma de código aberto de ponta a ponta para aprendizado de máquina. Ele possui um ecossistema abrangente e flexível de ferramentas, bibliotecas e recursos da comunidade”, fato este que permite aos pesquisadores a promoção do estado-da-arte em *machine learning*

(ML), e aos desenvolvedores a criarem e implementarem facilmente aplicativos baseados em ML.

Os pilares para a solidificação da plataforma TensorFlow se dá por três principais motivos. O primeiro, a construção fácil de modelos de ML, facilitando inclusive o seu treinamento, utilizando-se de APIs de alto nível, como o Keras, que facilita a interação do modelo e a sua depuração. O segundo ponto é sobre a possibilidade de utilização dos modelos tanto na nuvem, como no local, navegador ou dispositivo móvel, independentemente do idioma usado, o que nos leva ao terceiro ponto, da agilidade, como o TensorFlow possui alta flexibilidade e arquitetura simples, o mesmo possibilita que novas ideias tenham velocidade para sair do conceito e se tornarem código, com modelos avançados e prontos para a utilização. (TENSORFLOW, 2019).

Como a aplicação realizada por Sang e Di Pierro (2018) no mercado financeiro, que utilizando-se de uma rede neural com memória de longo prazo para aprender e aprimorar algoritmos de negociação, mostraram que a estratégia de previsão através dessa rede neural comparada a análise técnica tradicional, tem maior eficiência.

Outra aplicação, de Vázquez-Canteli et al. (2019) na área de energia e cidades inteligentes, traz a utilização do TensorFlow através da API Keras em conjunto com outro software específico para simulação de gasto de energia elétrica, o CitySim, criando um ambiente que permite aos pesquisadores encontrar novos algoritmos para controle de aprendizado em energia de cidades inteligentes. Esta aplicação demonstra a flexibilidade e interatividade do TensorFlow e sua principal API, o Keras.

2.7 API KERAS

Keras é uma Interface de Programação de Aplicações (API) de aprendizado profundo (*deep learning*) escrita em Python, rodando sobre a plataforma de aprendizado de máquina TensorFlow. Foi desenvolvido com o objetivo de permitir experimentação rápida. Ser capaz de passar da ideia para o resultado o mais rápido possível.

Keras é a API de alto nível do TensorFlow 2.0: uma interface acessível e altamente produtiva para resolver problemas de aprendizado de máquina, com foco no moderno aprendizado profundo. Ele fornece abstrações e blocos de construção

essenciais para o desenvolvimento e envio de soluções de aprendizado de máquina com alta velocidade de iteração.

O Keras capacita engenheiros e pesquisadores a aproveitar ao máximo os recursos de escalabilidade e plataforma cruzada do TensorFlow 2.0: você pode executar o Keras em TPU ou em grandes grupos de GPUs e pode exportar seus modelos Keras para rodar no navegador ou em um celular dispositivo (KERAS,2020).

2.8 LINGUAGEM DE PROGRAMAÇÃO PYTHON

Python é uma linguagem de programação de alto nível, interpretada, de script, imperativa, orientada a objetos, funcional, de tipagem dinâmica e forte. Foi lançada por Guido van Rossum em 1991. Atualmente possui um modelo de desenvolvimento comunitário, aberto e gerenciado pela organização sem fins lucrativos Python Software Foundation. Apesar de várias partes da linguagem possuírem padrões e especificações formais, a linguagem como um todo não é formalmente especificada. O padrão de facto é a implementação CPython.

A linguagem foi projetada com a filosofia de enfatizar a importância do esforço do programador sobre o esforço computacional. Prioriza a legibilidade do código sobre a velocidade ou expressividade. Combina uma sintaxe concisa e clara com os recursos poderosos de sua biblioteca padrão e por módulos e frameworks desenvolvidos por terceiros.

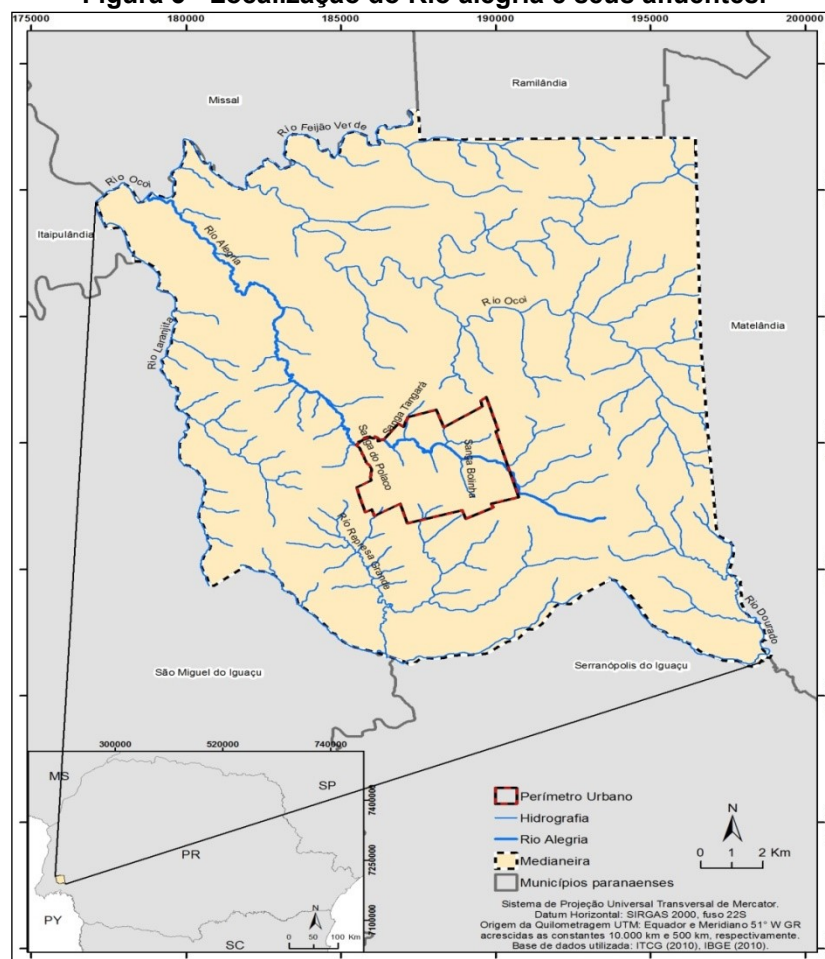
Python é uma linguagem de propósito geral, multiparadigma, suporta o paradigma orientado a objetos, imperativo, funcional e procedural. Possui tipagem dinâmica e uma de suas principais características é permitir a fácil leitura do código e exigir poucas linhas de código se comparado ao mesmo programa em outras linguagens. Devido às suas características, ela é principalmente utilizada para processamento de textos, dados científicos e criação de CGIs para páginas dinâmicas para a web. Foi considerada pelo público a 3ª linguagem "mais amada", de acordo com uma pesquisa conduzida pelo site Stack Overflow em 2018, e está entre as 5 linguagens mais populares, de acordo com uma pesquisa conduzida pela RedMonk.

3 PROCEDIMENTOS METODOLÓGICOS DA PESQUISA

3.1 ÁREA DE ESTUDO

O estudo foi realizado no Rio Alegria, no qual tem suas nascentes localizadas na área rural, e corta toda a área urbana do município de Medianeira no estado do Paraná, com uma extensão aproximada de 30,6 km. É um afluente do Rio Ocoy e está inserido na Bacia Hidrográfica do Rio Paraná 3, recebe água de seus afluentes, a Sanga Magnólia, a Sanga Manduri e a Sanga Maguari (SCHUTZ, 2014).

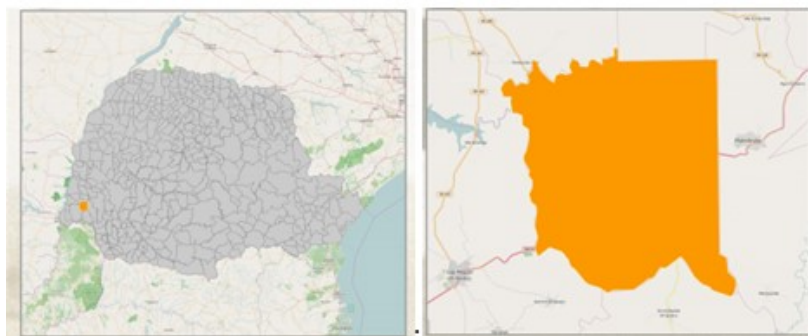
Figura 5 - Localização do Rio Alegria e seus afluentes.



Fonte: (URBANO, 2014)

A população estimada em 2020 era de 46.574 mil habitantes. A área da unidade territorial é de 328,732 km², com uma densidade demográfica de 127,21 hab/km².

Figura 6 - Malha municipal simplificada



Fonte: IBGE Malha municipal simplificada - 2020.

O clima de Medianeira é subtropical úmido, com temperatura média anual de 19 °C. O mês mais quente é dezembro com máxima pouco superior a 30°C e o mais frio julho com mínima próxima dos 8°C. As chuvas são abundantes durante o ano todo, não havendo a ocorrência de uma estação seca. O índice pluviométrico do município é de 1,920 mm/ano, sendo o mês mais chuvoso outubro e agosto o menos chuvoso (IBGE, 2020).

3.2 DADOS DE COLETA

Os dados de coleta da qualidade da água foram utilizados da tese de doutorado de Schutz 2014.

Para a obtenção dos dados, Schutz (2014) realizou 22 coletas com intervalo de 15 dias entre elas. Em cada um dos 6 pontos amostrados, foram coletados 03 litros de água, sendo 02 deles destinados para a determinação da demanda bioquímica de oxigênio (DBO5). As amostras foram enviadas para o Laboratório de Análises Físico-Químicas do Instituto Ambiental do Paraná (IAP), localizado na cidade de Toledo, PR.

Nas mesmas datas, foram coletadas amostras do efluente lançado, e avaliado DBO5, pelo laboratório da indústria responsável pelo lançamento.

Figura 7 - Pontos de amostragem e coleta de água no Rio Alegria



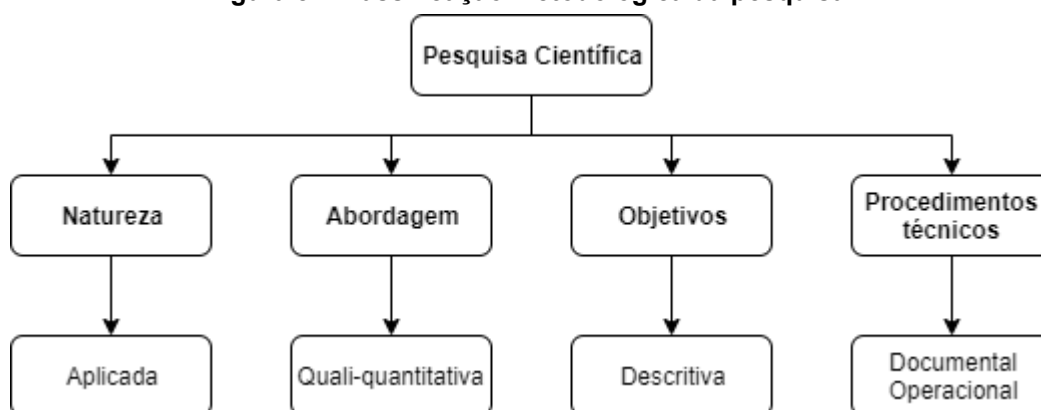
Fonte: (SCHUTZ, 2014).

O ponto 01 localiza-se à montante de uma importante indústria frigorífica da região, o ponto 02 localiza-se exatamente na calha de despejo do efluente do frigorífico, os pontos 03, 04, 05 e 06, localizam-se à jusante do ponto de despejo dos efluentes da referida indústria.

3.3 TIPO DE PESQUISA OU TÉCNICAS DE PESQUISA

Para o presente estudo realizou-se uma pesquisa científica conforme Figura 8.

Figura 8 - Classificação metodológica da pesquisa.



Fonte: Autoria Própria.

Quanto à natureza da pesquisa, este trabalho se classifica como do tipo aplicada, buscando a utilização dos resultados de forma imediata para solucionar problemas específicos (TURRIONI; MELLO, 2012).

Quanto ao tipo de abordagem, é classificado como pesquisa quantitativa, pois quantificará informações para posterior classificação e análise. A quantitativa preocupa-se em medir quantidade, frequência e intensidade, e analisar as relações causais entre as variáveis.

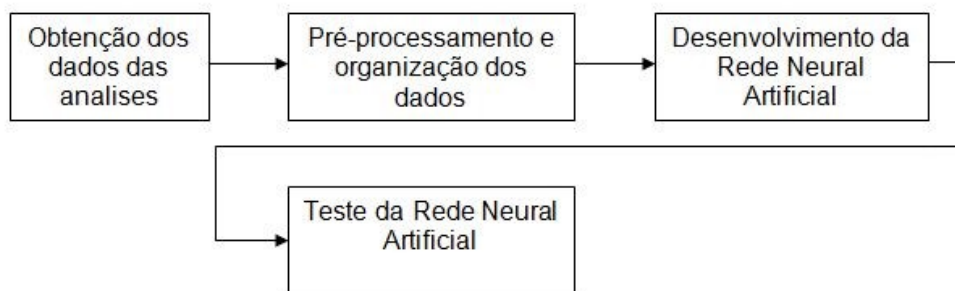
Os principais tipos de pesquisa quantitativa são a survey (levantamento), a correlacional, a causal-comparativa e a experimental. Reconhecem-se como principais formas de abordagem qualitativa a pesquisa-ação, a pesquisa histórica, o estudo de caso, o focus group, a etnografia e a grounded theory (VERGARA, 2005).

Quanto aos objetivos, a pesquisa é classificada como exploratória, que, segundo Cervo; Bervian; Da Silva (2007), busca a familiarização com os fenômenos observados, gerando novas ideias e percepções. Os mesmos ainda mencionam que a elaboração de hipóteses não se faz necessário, atendo-se apenas a elaboração de objetivos, que terá por função nortear a busca de informações para descrever precisamente o fenômeno e as relações entre seus componentes

O presente estudo, segundo Cardoso (2011) classifica-se como uma pesquisa operacional por apresentar uma programação otimizada de atividades ou recursos, fornecendo um conjunto de procedimentos e métodos quantitativos para tratar de forma sistematizada problemas que envolvam a utilização de recursos escassos. Para apoiar a tomada de decisão, a PO busca a solução de problemas que podem ser representados por modelos matemáticos.

3.4 FERRAMENTAS UTILIZADAS

Para o desenvolvimento deste estudo, foi necessário a utilização de um banco de dados, composto por dados de análise de qualidade da água a fim de caracterizar a Bacia do Rio Alegria no município de Medianeira - PR. O fluxo de procedimentos adotados no presente trabalho está conforme Figura 9:

Figura 9 - Fluxograma metodológico do estudo

Fonte: Autoria própria.

De início o principal objetivo foi o de realizar o mapeamento dos processos de análise e qualidade da água, a fim de entender como o mesmo funciona, encontrando uma melhor disposição dos dados para uma inserção da inteligência artificial que foi desenvolvida.

Posteriormente às análises supracitadas, o objetivo foi selecionar os dados para iniciar o desenvolvimento, treinamento e teste da RNA, sobre os dados passados, sendo que nesta sequência, o desenvolvimento vem a ser efetivamente a criação da RNA, tendo o treinamento e teste como partes desse passo.

Nesta sessão descreveu-se as ferramentas utilizadas para desenvolvimento do estudo.

A plataforma Anaconda foi usada como um gerenciador de pacotes para ciência de dados e IA. Com ela foi possível criar ambientes com versões específicas de cada pacote que irá utilizar em seu algoritmo, garantindo a compatibilidade entre eles.

O Spyder foi usado como ambiente de desenvolvimento. com a utilização de recursos avançados de edição, análise e depuração de dados, acelerando a implementação.

O Tensorflow foi usado como biblioteca, pois possui um ecossistema abrangente e flexível de ferramentas, e recursos da comunidade.

Keras, como API de aprendizado profundo (*deep learning*) escrita em Python, rodando sobre a plataforma de aprendizado de máquina TensorFlow. Foi usado com o objetivo de permitir experimentação rápida.

Os dados das análises onde a RNA foi implementada e testadas estão dispostos no apêndice A.

4 RESULTADOS

Os dados de vazão e de qualidade da água e do efluente foram utilizados como entrada para o desenvolvimento e treinamento dos modelos baseados em Redes Neurais Artificiais (RNA) para caracterização do oxigênio dissolvido na água do Rio.

Foram desenvolvidos 5 modelos diferentes nomeados RNA1, RNA2, RNA3, RNA4 e RNA5 e aplicados inúmeros testes em cada modelo, variando-se a arquitetura da rede. As redes foram desenvolvidas com a validação cruzada para melhor avaliar a validade da rede neural na estimativa do oxigênio dissolvido.

Na primeira versão nomeada RNA1, com 10 entradas sendo elas: DBO, DQO, pH, CE, To, Nitrito, Nitrato, Nitrogênio amoniacal, vazão e uma saída :OD apresentava arquitetura 10:10:1, com apenas uma camada oculta de neurônios e 1000 épocas. Os dados foram agrupados de forma que as informações das análises realizadas com o efluente figurassem na rede como um ponto a mais de coleta. Nesta versão foram incluídos dados referentes a pH, CE, T, Nitrito, Nitrato, e OD.do efluente, fornecidos pelo laboratório da indústria geradora do efluente, além dos anteriormente previstos a figura Figura 10 mostra o modelo.

Figura 10 - Modelo RNA1

```
def build_model():
    model = keras.Sequential([
        layers.Dense(10, input_shape=[len(train_dataset.keys())]),
        layers.Dense(10),
        layers.Dense(1)
    ])
    optimizer = tf.keras.optimizers.RMSprop(0.001)

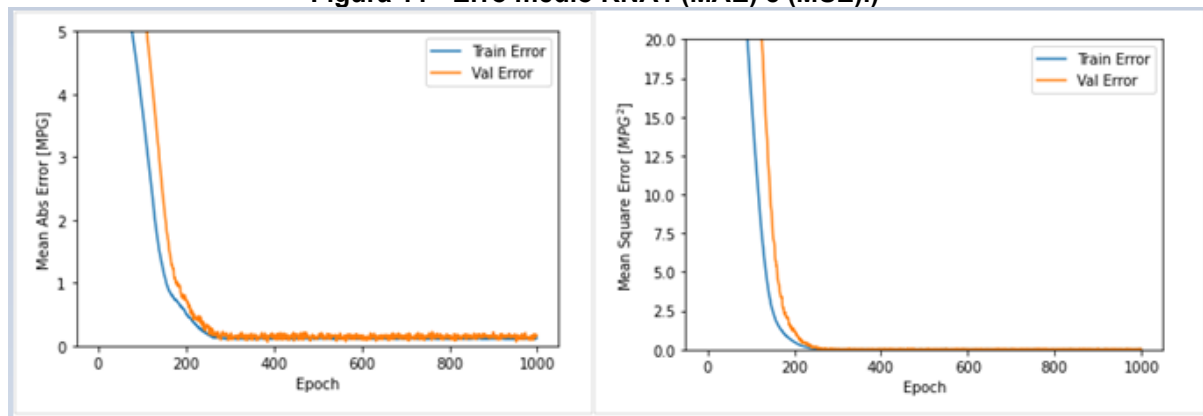
    model.compile(loss='mse',
                  optimizer=optimizer,
                  metrics=['mae', 'mse'])
    return model
```

Fonte: Autoria própria.

O erro médio absoluto (MAE) é uma medida de erros entre observações pareadas que expressam o mesmo fenômeno. Exemplos de Y versus X incluem

comparações de predito versus observado. É, portanto, uma média aritmética dos erros absolutos, conforme Figura 11.

Figura 11 - Erro médio RNA1 (MAE) e (MSE).)



Fonte: Autoria própria.

O erro quadrático médio (MSE) ou desvio quadrático médio (MSD) de um estimador (de um procedimento para estimar uma quantidade não observada) mede a média dos quadrados dos erros, isto é, a diferença quadrática média entre os valores estimados e o valor real. MSE é uma função de risco, correspondendo ao valor esperado da perda de erro quadrática. O fato de o MSE ser quase sempre estritamente positivo (e não zero) é devido à aleatoriedade ou porque o estimador não leva em conta informações que poderiam produzir uma estimativa mais precisa.

Os valores de métricas dos testes foram:

MAE (Mean Absolute Error) = 0.1772786662.

MSE (Mean Squared Error) = 0.0419589263

RMSE = 0.2048387812.

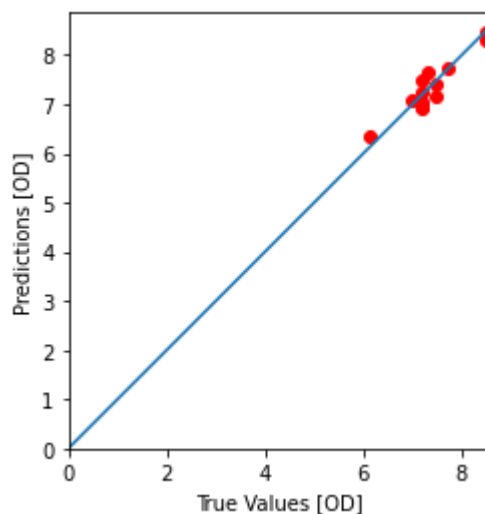
MAPE = 2.4512672800%.

O erro percentual absoluto médio (MAPE) é comumente usado como uma função de perda para problemas de regressão e na avaliação de modelos, devido à sua interpretação muito intuitiva em termos de erro relativo.

O erro quadrático médio (RMSE) é uma regra de pontuação quadrática que também mede a magnitude média do erro. É a raiz quadrada da média das diferenças quadradas entre a previsão e a observação real.

. Na figura 12 pode-se observar a comparação entre os dados conhecidos e os dados simulados pela rede.

Figura 12 - Valores de OD reais x OD Previstos RNA1.



Fonte: Autoria própria.

Com o intuito de buscar os melhores resultados para a RNA, buscou-se realizar testes, variando diversos parâmetros, como: *EarlyStopping callback* que testa a condição do treinamento a cada *epoch* se um grupo de *epochs* decorre sem mostrar melhoras, o treinamento irá parar automaticamente conforme Figura 13.

Figura 13 - Treinamento do modelo com *EarlyStopping*

```

model = build_model()

# O parâmetro patience é o quantidade de epochs para checar as melhoras
early_stop = keras.callbacks.EarlyStopping(monitor='val_loss', patience=20)

history = model.fit(normed_train_data, train_labels, epochs=EPOCHS,
                    validation_split = 0.2, verbose=0, callbacks=[early_stop, PrintDot()])

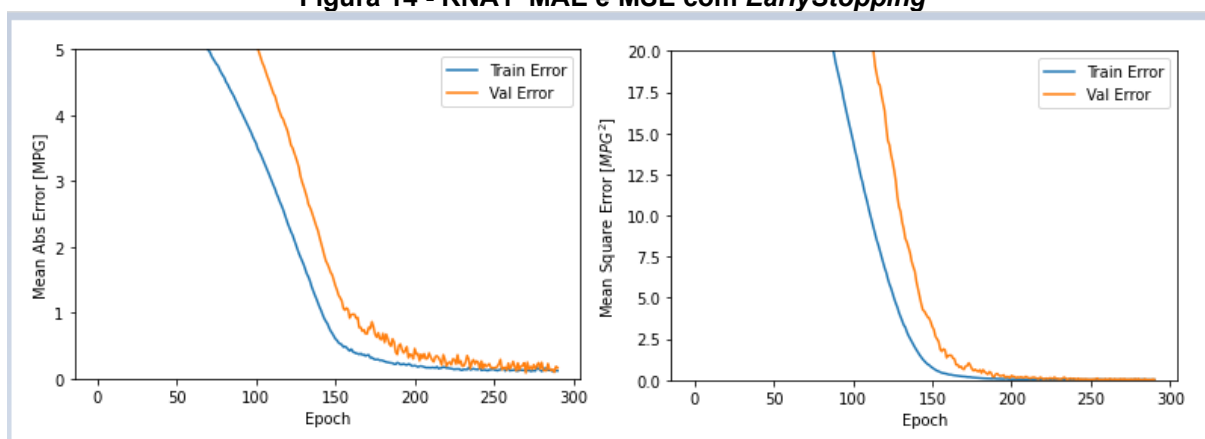
plot_history(history)

```

Fonte: Autoria própria.

A Figura 14 mostra o momento que o treinamento para assim que um grupo de *epochs* estabiliza sem mostrar melhoras.

Figura 14 - RNA1 MAE e MSE com *EarlyStopping*

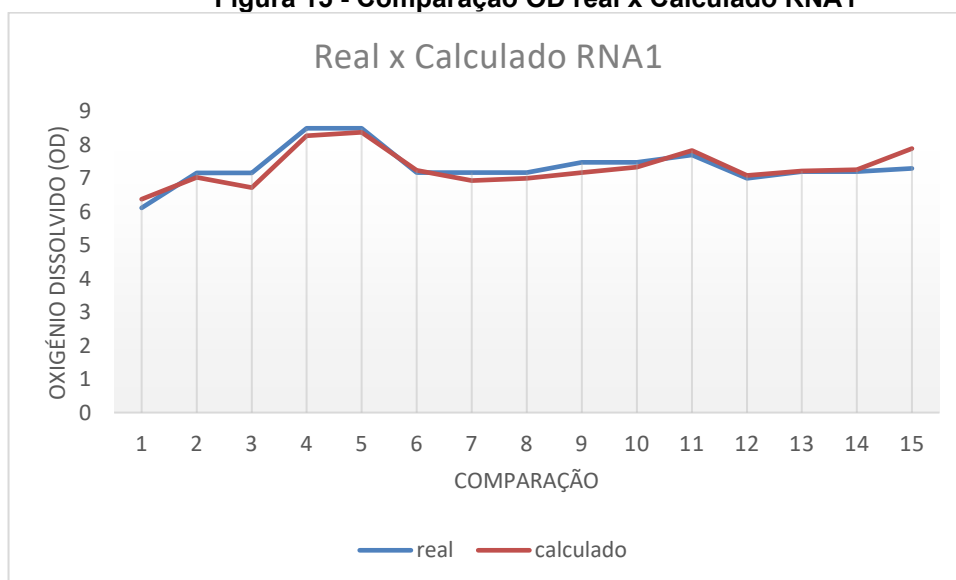


Fonte: Autoria própria.

É possível notar que com quase 300 *epochs* o modelo encerra o treinamento pois tem se a estabilização da melhora.

Por fim o erro médio ficou em 0.1772786662 a acurácia geral ficou em 0.90495 com desvio padrão de 0.01837, o tempo médio foi de menos de 2 minutos de processamento. Na Figura 15 pode-se observar a comparação entre os dados conhecidos e os dados simulados pela rede.

Figura 15 - Comparação OD real x Calculado RNA1

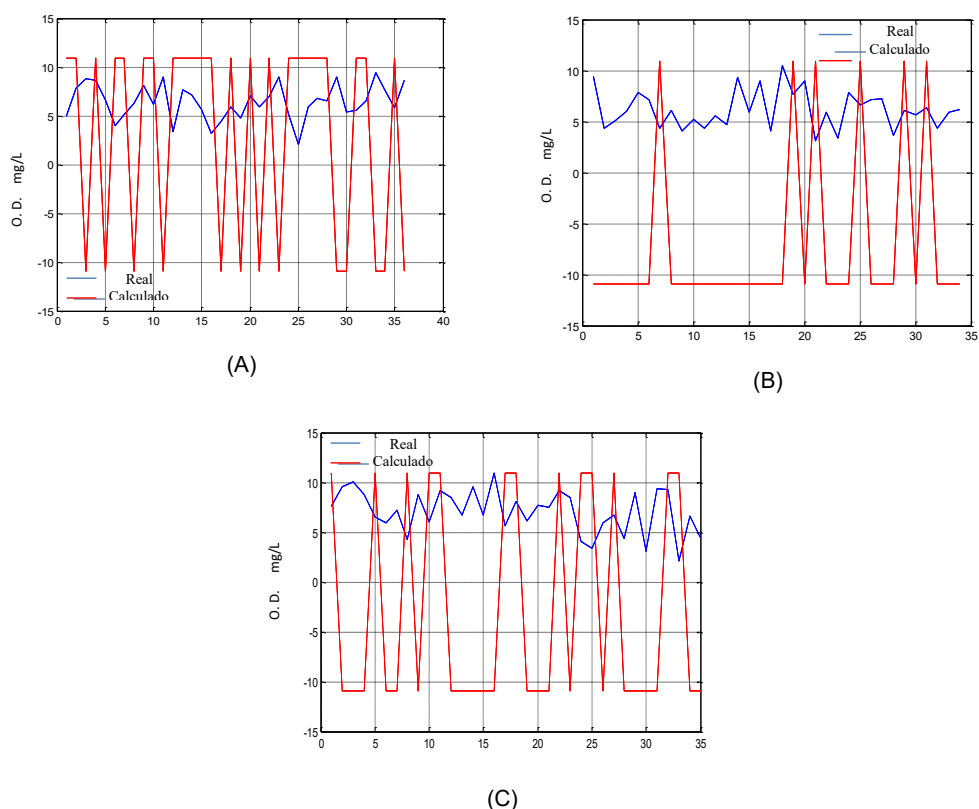


Fonte: Autoria própria.

No trabalho desenvolvido por Schutz(2014) a RNA1 teve o tempo médio de processamento de 12 horas, o modelo gerou um erro médio entre os valores de Oxigênio dissolvido conhecidos e os valores ajustados de 398 % e um coeficiente de

determinação 0,0002. Na Figura 16 pode-se observar a comparação entre os dados conhecidos e os dados simulados pela rede desenvolvida por Schutz(2014).

Figura 16 - Gráficos gerados na validação cruzada do primeiro modelo RNA1 Oxigênio dissolvido real e calcula



Fonte: Schutz(2014).

Foi desenvolvido mais um modelo nomeado RNA2 com 16 entradas, agora considerando os valores em duplicata de DBO, DQO do efluente como duas entradas distintas. Ou seja: da água foram dados de entrada: DBO, DQO, pH, CE, T, Nitrito, Nitrato, Nitrogênio amoniacal, vazão. E do efluente foram dados de entrada: DBO, DQO, DBO, DQO Nitrogênio amoniacal, e vazão) e uma saída (OD).

Figura 17 - Modelo RNA2

```
def build_model():
    model = keras.Sequential([
        layers.Dense(16, input_shape=[len(train_dataset.keys())]),
        layers.Dense(32),
        layers.Dense(32),
        layers.Dense(1)
    ])

    optimizer = tf.keras.optimizers.RMSprop(0.001)

    model.compile(loss='mse',
                  optimizer=optimizer,
                  metrics=['mae', 'mse'])

    return model
```

Fonte: Autoria própria.

Os valores de métricas dos testes foram:

MAE = 0.1512660166.

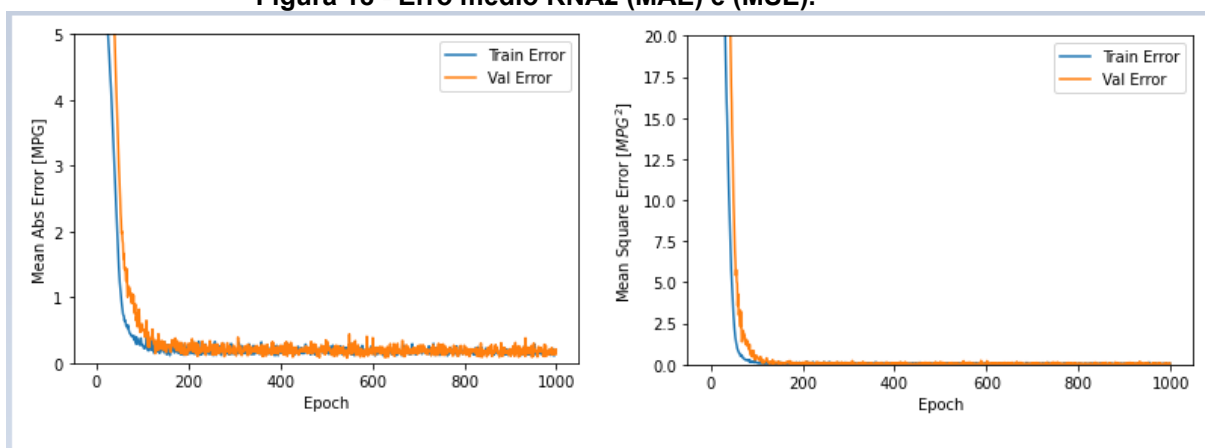
MSE = 0.0421406941.

RMSE = 0.2052819867.

MAPE = 2.0701983025%

Na Figura 18 é possível observar o MAE e o MSE em um período de 1000 *epochs*.

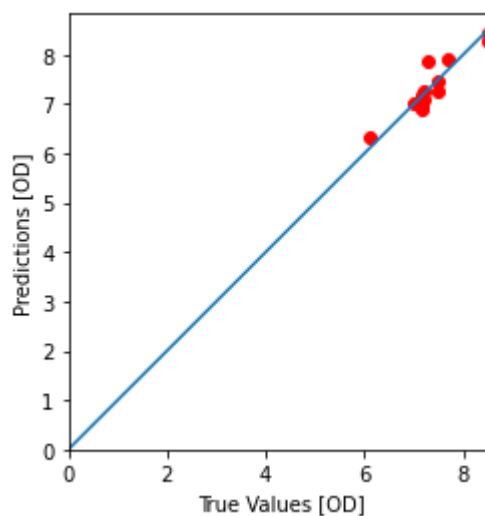
Figura 18 - Erro médio RNA2 (MAE) e (MSE).



Fonte: Autoria própria.

Na Figura 19 pode-se observar a comparação entre os dados conhecidos e os dados simulados pela rede.

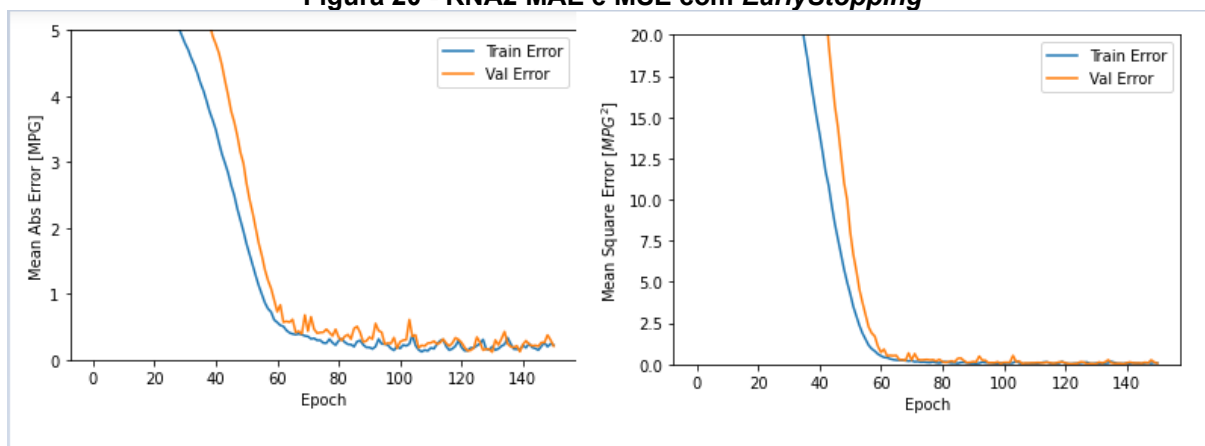
Figura 19 - Valores de OD reais x OD Previstos RNA2.



Fonte: Autoria própria.

A Figura 20 mostra o momento que o treinamento para assim que um grupo de *epochs* estabiliza sem mostrar melhoras, foi feito o mesmo tipo de configuração de *EarlyStopping* conforme mostrado na Figura 13.

Figura 20 - RNA2 MAE e MSE com *EarlyStopping*

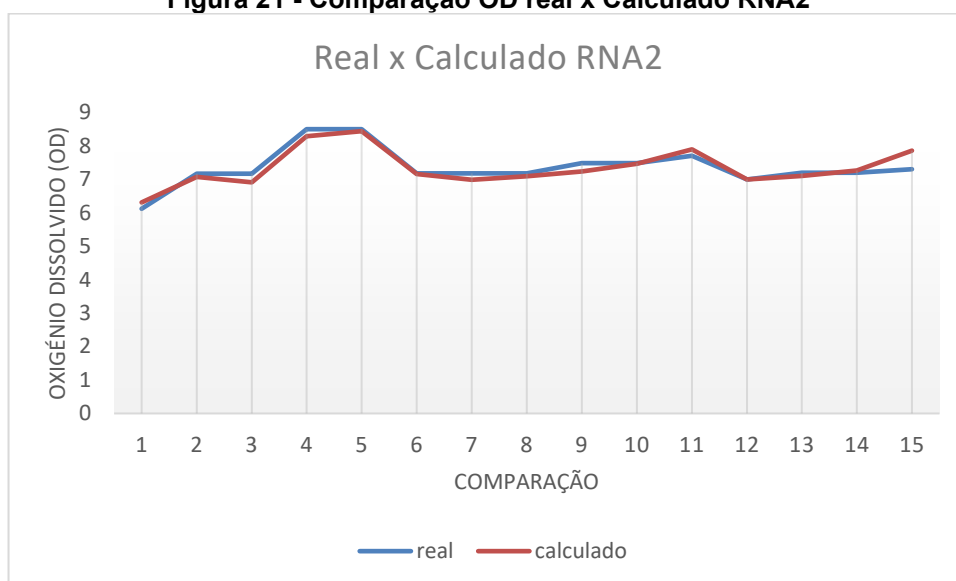


Fonte: Autoria própria.

É possível notar que com 160 *epochs* o modelo encerra o treinamento pois tem se a estabilização da melhora.

Por fim o erro médio ficou em 0.1512660166 a acurácia geral ficou em 0.89625 com desvio padrão de 0.01811, o tempo médio foi de menos de 2 minutos de processamento. Na Figura 21 pode-se observar a comparação entre os dados conhecidos e os dados simulados pela rede.

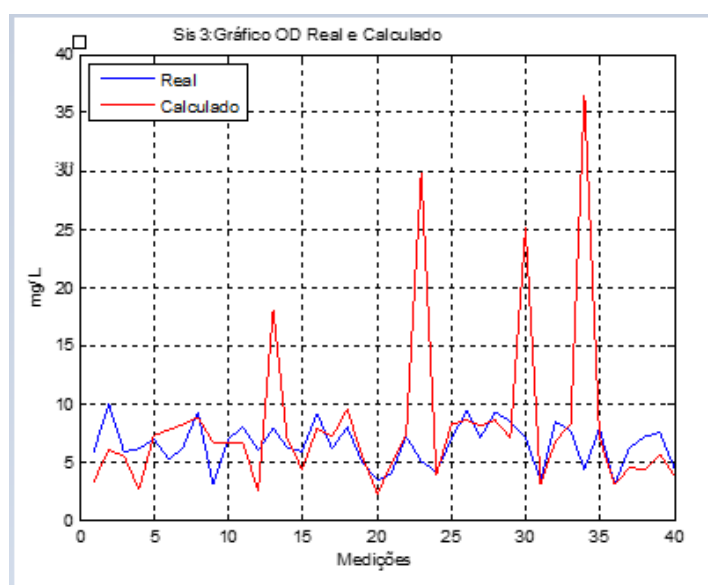
Figura 21 - Comparação OD real x Calculado RNA2



Fonte: Autoria própria.

No trabalho desenvolvido por Schutz(2014) a RNA2 teve o tempo médio de processamento de 18 horas, o modelo gerou um erro médio entre os valores de Oxigênio dissolvido conhecidos e os valores ajustados de 32,5 % e um coeficiente de determinação 0,11. Na **Figura 22** pode-se observar a comparação entre os dados conhecidos e os dados simulados pela rede desenvolvida por Schutz(2014).

Figura 22 - Comparação entre os dados conhecidos e os dados simulados pela rede RNA2.



Fonte: Schutz(2014).

Foi desenvolvido mais um modelo nomeado de RNA3 uma configuração com três camadas ocultas 14:28:28:14:1 e com 14 entradas, utilizando-se os valores médios das análises em duplicata de DBO5, DQO do efluente e uma saída.

Figura 23 - Modelo RNA3

```
def build_model():
    model = keras.Sequential([
        layers.Dense(14, input_shape=[len(train_dataset.keys())]),
        layers.Dense(28),
        layers.Dense(28),
        layers.Dense(14),
        layers.Dense(1)
    ])

    optimizer = tf.keras.optimizers.RMSprop(0.001)

    model.compile(loss='mse',
                  optimizer=optimizer,
                  metrics=['mae', 'mse'])

    return model
```

Fonte: Autoria própria.

Os valores de métricas dos testes foram:

MAE = 0.3774264018.

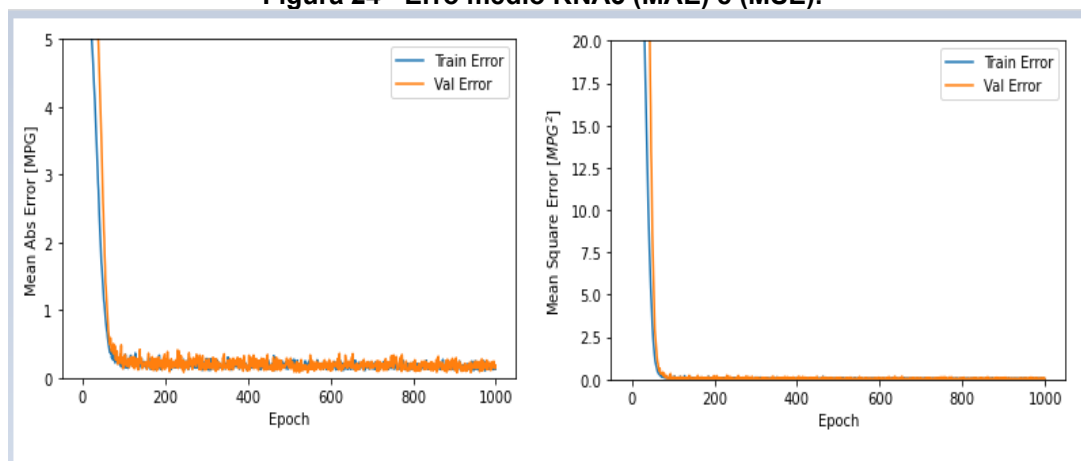
MSE = 0.2438415254.

RMSE = 0.4938031241.

MAPE = 5.1248332527%.

Na Figura 24 é possível observar o MAE e o MSE em um período de 1000 *epochs*.

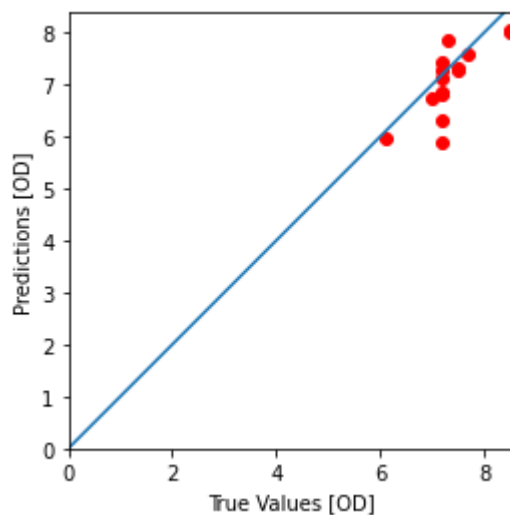
Figura 24 - Erro médio RNA3 (MAE) e (MSE).



Fonte: Autoria própria.

Na **Figura 25** pode-se observar a comparação entre os dados conhecidos e os dados simulados pela rede.

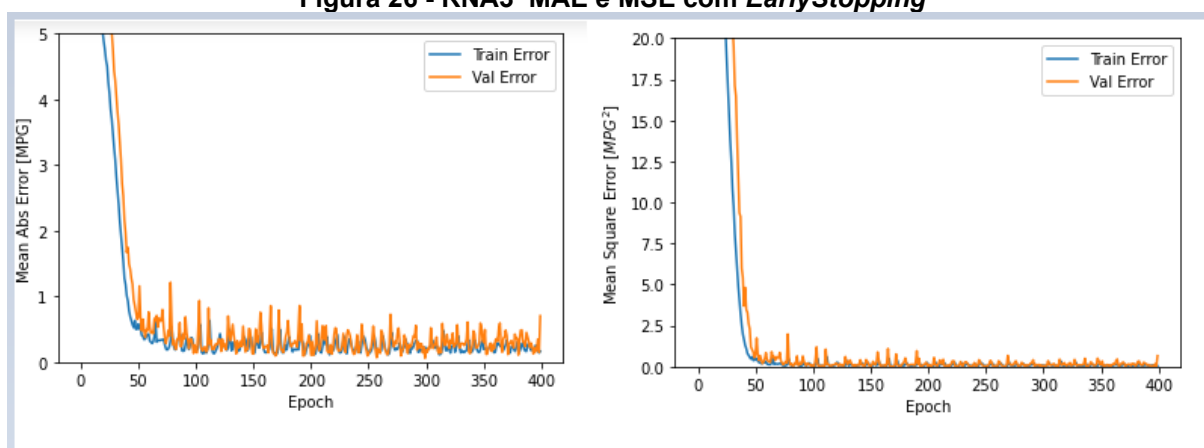
Figura 25 - Valores de OD reais x OD Previstos RNA3.



Fonte: Autoria própria.

A **Figura 26** mostra o momento que o treinamento para assim que um grupo de *epochs* estabiliza sem mostrar melhoras, foi feito o mesmo tipo de configuração de *EarlyStopping* conforme mostrado na **Figura 13**.

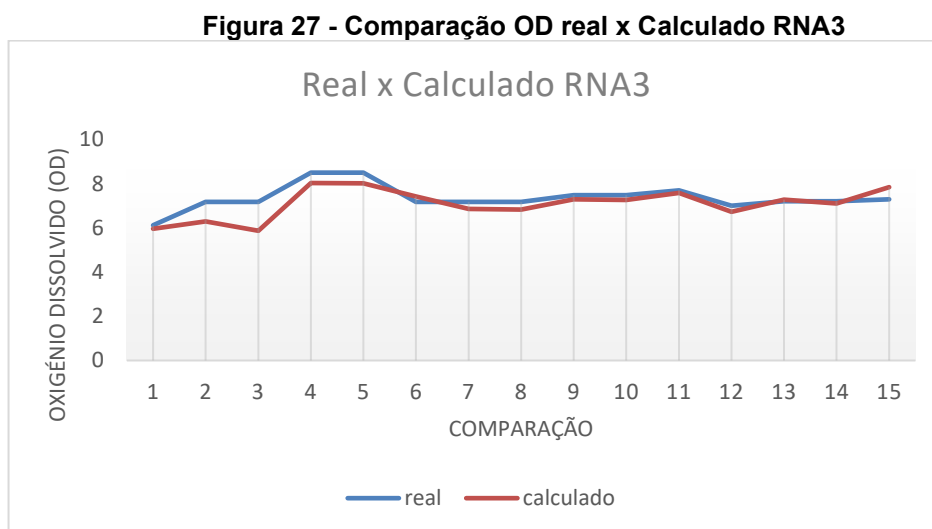
Figura 26 - RNA3 MAE e MSE com *EarlyStopping*



Fonte: Autoria própria.

É possível notar que com 400 *epochs* o modelo encerra o treinamento pois tem se a estabilização da melhora.

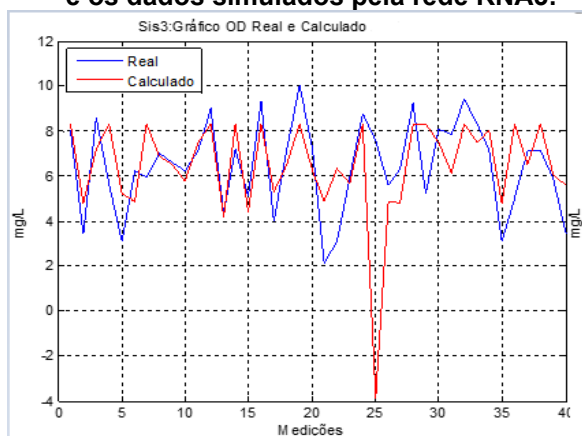
Por fim o erro médio ficou em 0.3774264018 a acurácia geral ficou em 0.87267 com desvio padrão de 0.03473, o tempo médio foi de menos de 2 minutos de processamento. Na Figura 27 pode-se observar a comparação entre os dados conhecidos e os dados simulados pela rede.



Fonte: Autoria própria.

No trabalho desenvolvido por Schutz (2014) a RNA3 teve o tempo médio de processamento de 18 horas, o modelo gerou um erro médio entre os valores de Oxigênio dissolvido conhecidos e os valores ajustados de 29,1 % e um coeficiente de determinação 0,061. Na Figura 28 pode-se observar a comparação entre os dados conhecidos e os dados simulados pela rede desenvolvida por Schutz (2014).

Figura 28 - Comparação entre os dados conhecidos e os dados simulados pela rede RNA3.



Fonte: Schutz (2014).

Desenvolveu-se um novo modelo nomeado RNA4 com 10 entradas e uma saída, sendo retirados os dados do efluente. Uma configuração com três camadas ocultas 10:20:20:10:1.

Figura 29 - Modelo RNA4

```
def build_model():
    model = keras.Sequential([
        layers.Dense(10, input_shape=[len(train_dataset.keys())]),
        layers.Dense(20),
        layers.Dense(20),
        layers.Dense(10),
        layers.Dense(1)
    ])

    optimizer = tf.keras.optimizers.RMSprop(0.001)

    model.compile(loss='mse',
                  optimizer=optimizer,
                  metrics=['mae', 'mse'])

    return model
```

Fonte: Autoria própria.

Os valores de métricas dos testes foram:

MAE = 0.2316159999.

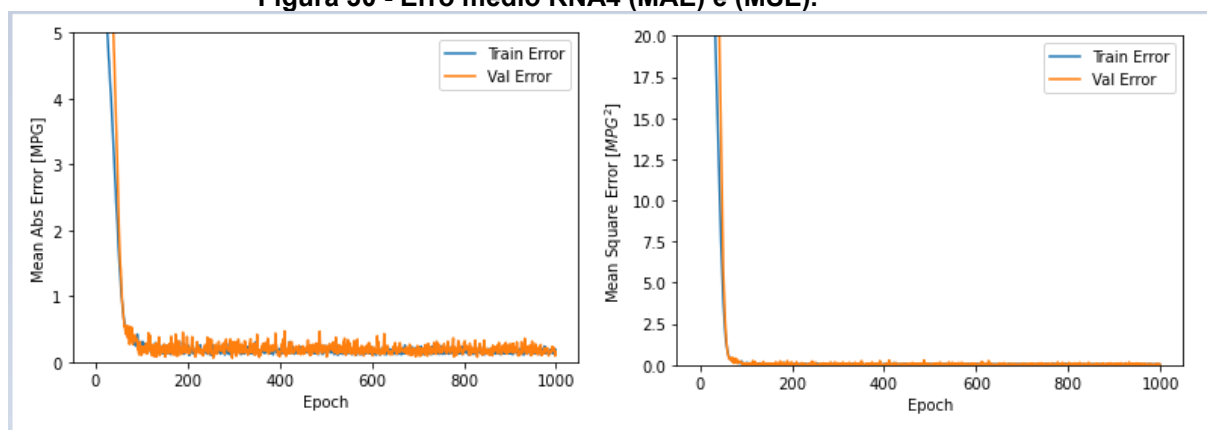
MSE = 0.0786245896.

RMSE = 0.2804007661.

MAPE = 3.1979167502%.

Na Figura 30 é possível observar o MAE e o MSE em um período de 1000 *epochs*.

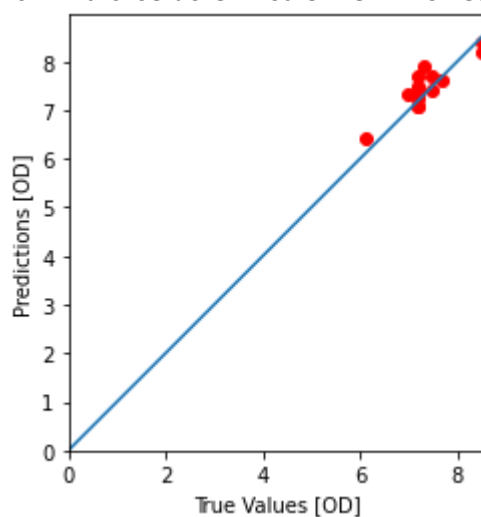
Figura 30 - Erro médio RNA4 (MAE) e (MSE).



Fonte: Autoria própria.

Na Figura 31 pode-se observar a comparação entre os dados conhecidos e os dados simulados pela rede.

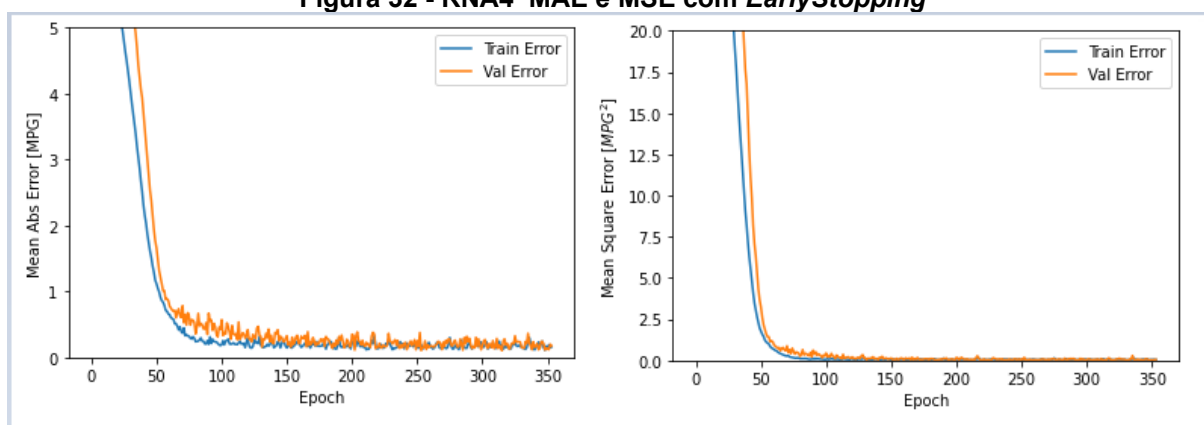
Figura 31 - Valores de OD reais x OD Previstos RNA4.



Fonte: Autoria própria.

A Figura 32 mostra o momento que o treinamento para assim que um grupo de *epochs* estabiliza sem mostrar melhoras, foi feito o mesmo tipo de configuração de *EarlyStopping* conforme mostrado na Figura 13.

Figura 32 - RNA4 MAE e MSE com *EarlyStopping*

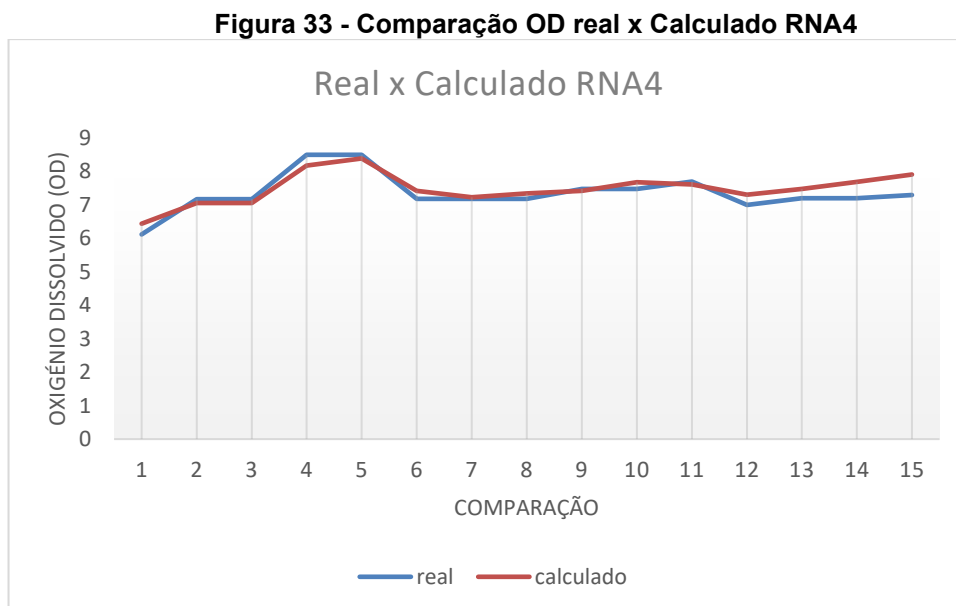


Fonte: Autoria própria.

É possível notar que com 350 *epochs* o modelo encerra o treinamento pois tem se a estabilização da melhora.

Por fim o erro médio ficou em 0.2316159999 a acurácia geral ficou em 0.87490 com desvio padrão de 0.02966, o tempo médio foi de menos de 2 minutos de

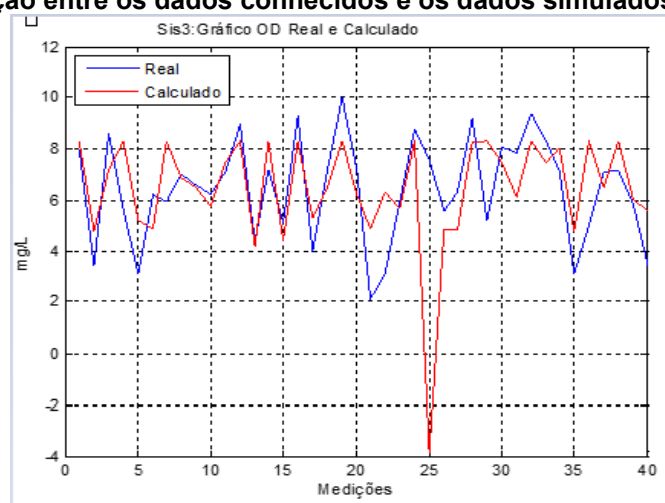
processamento. Na Figura 33 pode-se observar a comparação entre os dados conhecidos e os dados simulados pela rede.



Fonte: Autoria própria.

No trabalho desenvolvido por Schutz (2014) a RNA4 teve o tempo médio de processamento de 12 horas, o modelo gerou um erro médio entre os valores de Oxigênio dissolvido conhecidos e os valores ajustados de 15,5% e um coeficiente de determinação 0,60. Na Figura 34 pode-se observar a comparação entre os dados conhecidos e os dados simulados pela rede desenvolvida por Schutz (2014).

Figura 34 - Comparação entre os dados conhecidos e os dados simulados pela rede RNA4



Fonte: Schutz (2014).

Foi criada uma nova rede RNA5, com 10 entradas e uma saída foi utilizada uma configuração com três camadas ocultas 10:10:10:10:1.

Figura 35 - Modelo RNA5

```
def build_model():
    model = keras.Sequential([
        layers.Dense(10, input_shape=[len(train_dataset.keys())]),
        layers.Dense(10),
        layers.Dense(10),
        layers.Dense(10),
        layers.Dense(1)
    ])

    optimizer = tf.keras.optimizers.RMSprop(0.001)

    model.compile(loss='mse',
                  optimizer=optimizer,
                  metrics=['mae', 'mse'])

    return model
```

Fonte: Autoria própria.

Os valores de métricas dos testes foram:

MAE = 0.1985926857.

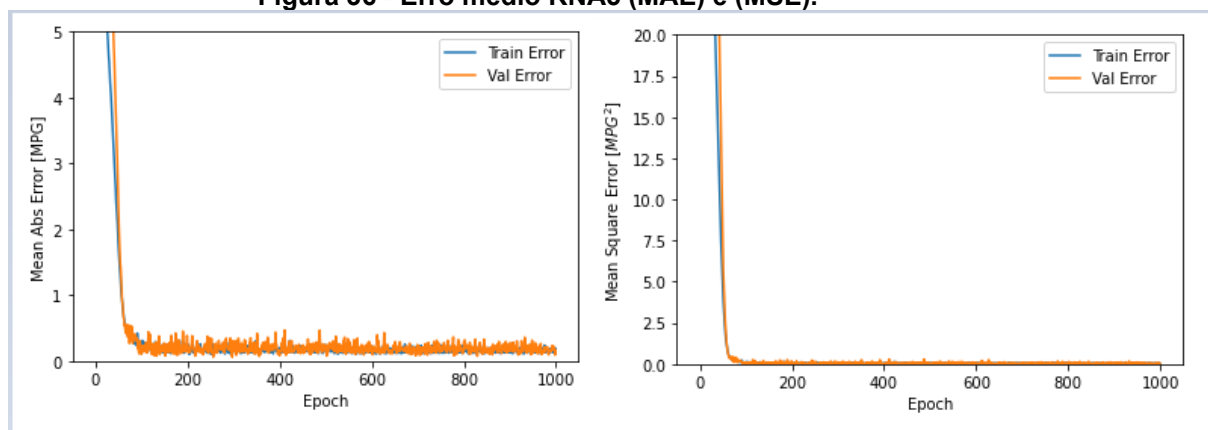
MSE = 0.0624495506.

RMSE = 0.2498990808.

MAPE = 2.7231984048%.

Na Figura 36 é possível observar o MAE e o MSE em um período de 1000 *epochs*.

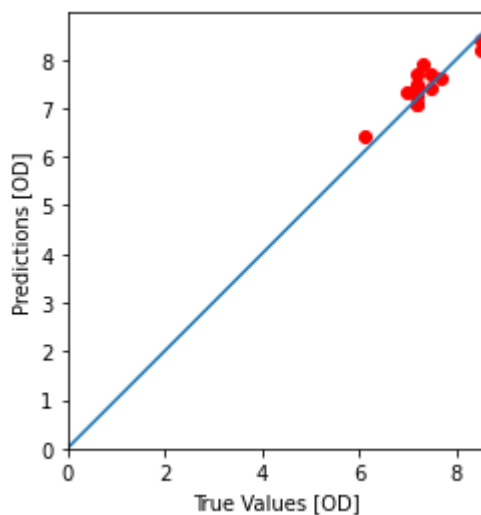
Figura 36 - Erro médio RNA5 (MAE) e (MSE).



Fonte: Autoria própria.

Na Figura 37 pode-se observar a comparação entre os dados conhecidos e os dados simulados pela rede.

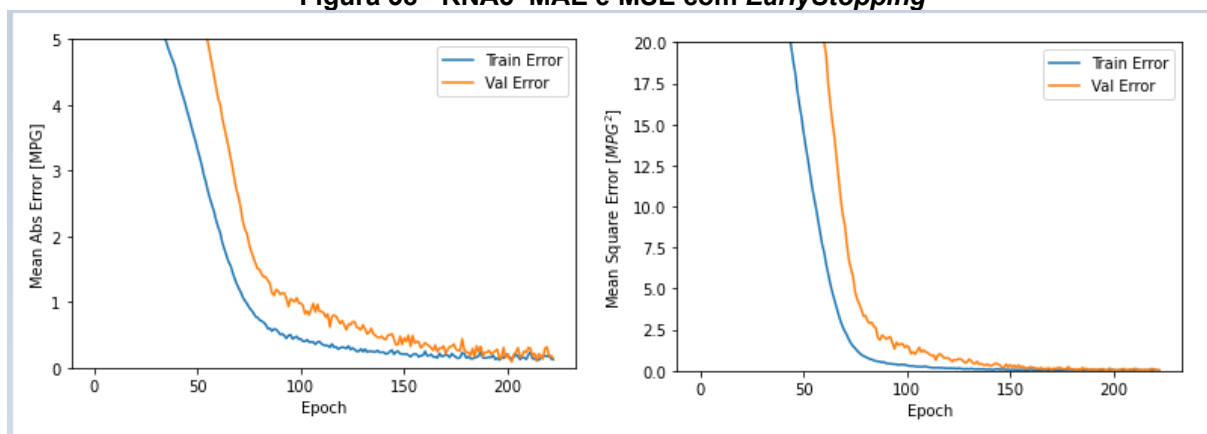
Figura 37 - Valores de OD reais x OD Previstos RNA5.



Fonte: Autoria própria.

A Figura 38 mostra o momento que o treinamento para assim que um grupo de *epochs* estabiliza sem mostrar melhoras, foi feito o mesmo tipo de configuração de *EarlyStopping* conforme mostrado na Figura 13.

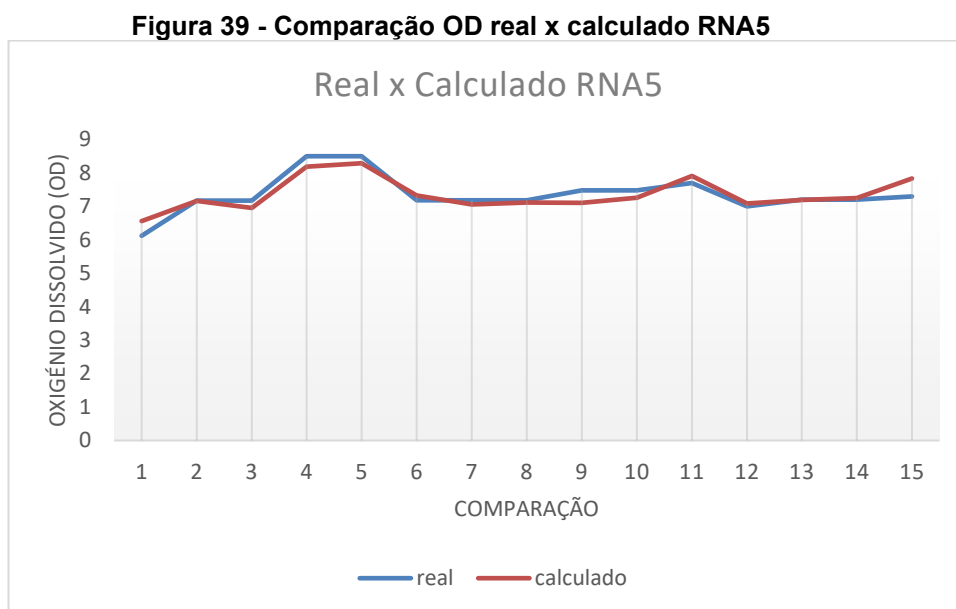
Figura 38 - RNA5 MAE e MSE com *EarlyStopping*



Fonte: Autoria própria.

É possível notar que com 250 *epochs* o modelo encerra o treinamento pois tem se a estabilização da melhora.

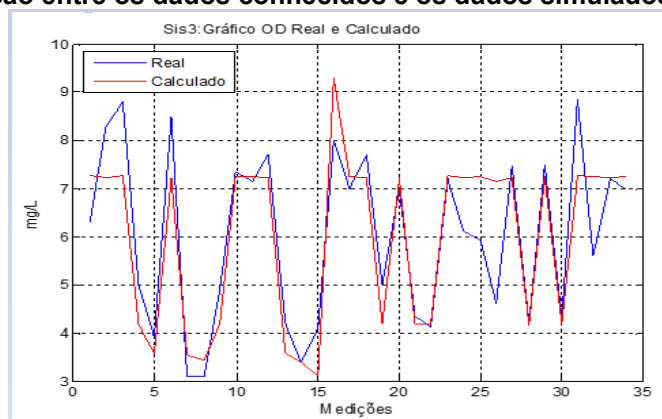
Por fim o erro médio ficou em 0.1985926857 a acurácia geral ficou em 0.90303 com desvio padrão de 0.02375, o tempo médio foi de menos de 2 minutos de processamento. Na Figura 39 pode-se observar a comparação entre os dados conhecidos e os dados simulados pela rede.



Fonte: Autoria própria.

No trabalho desenvolvido por Schutz (2014) a RNA5 teve o tempo médio de processamento de 12 horas, o modelo gerou um erro médio entre os valores de Oxigênio dissolvido conhecidos e os valores ajustados de 14,5% e um coeficiente de determinação 0,77. Na Figura 40 pode-se observar a comparação entre os dados conhecidos e os dados simulados pela rede desenvolvida por Schutz (2014).

Figura 40 - Comparação entre os dados conhecidos e os dados simulados pela rede RNA5



Fonte: Schutz (2014).

Na Figura 40 foi possível perceber que a RNA5 do estudo desenvolvido por Schutz (2014) foi a que melhor apresentou resultados de simular o OD. Na Tabela 1 a seguir existe um comparativo dos resultados do presente estudo.

Tabela 1 – Comparativo de resultados entres as RNA's

	ARQUITETURA	MAE	MSE	RMSE	MAPE %	TEMPO	DESV.PAD.	ACURÁCIA
RNA1	10:10:01	0,177278666	0,041958926	0,204838781	2,45126728	2 min.	0,01837	0,90495
RNA2	16:32:32:1	0,151266017	0,042140694	0,205281987	2,07019830	2 min.	0,01811	0,89625
RNA3	14:28:28:14:1	0,377426402	0,243841525	0,493803124	5,12483325	2 min.	0,03473	0,87267
RNA4	10:20:20:10:1	0,231615999	0,078624589	0,280400766	3,19791675	2 min.	0,02966	0,8749
RNA5	10:10:10:10:1	0,198592686	0,062449551	0,249899081	2,72319840	2 min.	0,02375	0,90303

Se for considerada apenas a acurácia a RNA1 é a que melhor apresentou capacidade de simulação do OD em comparação com os dados coletados, entretanto se compararmos as demais métricas percebe-se que a RNA que teve o menor erro quadrático e o menor erro absoluto e uma acurácia de 0,89 foi a RNA2, isso significa que nestas condições tem-se uma rede que consegue simular melhor a autodepuração em rios, a RNA2 teve todos os parâmetros de qualidade da água coletados usados para treinamento da rede. Então percebe-se que em uma possível variação na entrada de efluentes a RNA mencionada conseguira prever o comportamento do rio, e simular quanto OD contem no curso d'água.

5 CONCLUSÃO

Considerando os resultados referentes as simulações realizadas com os modelos RNA1, RNA2, RNA3, RNA4 e RNA5 pode-se concluir que uma rede neural artificial pode ser empregada para prever o oxigênio dissolvido nas águas de um rio.

É possível dizer também que os modelos desenvolvidos com as novas tecnologias além de mais rápidos, tendem a ser mais precisos, além de fornecer uma forma mais simples de implementação com várias bibliotecas disponíveis no mercado e de forma gratuita.

Considerando os resultados, o modelo implementado pode ser uma ferramenta de grande valia, para a agroindústria e a indústria em geral, pois existem outorgas de uso das águas tanto para captação, quanto para despejo. Essas outorgas são condicionadas a manutenção e monitoramento da qualidade da água a montante e a jusante do rio.

Para trabalhos futuros, sugere-se treinar e validar o modelo com um banco de dados maior e com informações da qualidade de vários rios diferentes, validando a hipótese do modelo poder ser utilizado de uma forma geral no auxílio e monitoramento de cursos d'água. E também usar o processo de testar diferentes parâmetros da rede e ver quais os que oferecem melhores retornos, chamado de hiperparâmetros.

REFERÊNCIAS

- ANDRIOLO, M. et al, **Calibração do Modelo SWAT Para a Produção e Transporte de Sedimentos**. Simpósio Sobre Pesquisa e Médias Centrais Hidrelétricas, Belo Horizonte, 2008.
- ANA, Agência Nacional de Águas. Dados hidrológicos: **séries históricas**. Disponível em: <http://hidroweb.ana.gov.br>. Acesso em: 05 ago. 2016.
- BALDISSERA, G. C. **Aplicabilidade do modelo de simulação hidrológica SWAT (Soil and Water Assessment Tool), para a bacia hidrográfica do Rio Cuiabá/MT**. Cuiabá - MT. Dissertação (mestrado em Física e Meio Ambiente). Universidade Federal do Mato Grosso, 2005.
- BARRETO, J. M. Inteligência artificial – No limiar do século XXI. 3ª edição - Florianópolis – SC. O autor., 2005. 397 p.
- BOULOMYTIS, V. T. G. et al. WATERSHED SPATIAL DISCRETIZATION FOR THE ANALYSIS OF LAND USE CHANGE IN COASTAL REGIONS. **Boletim de Ciências Geodésicas**, [s.l.], v. 23, n. 1, p.101-114, mar. 2017. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s1982-21702017000100007>.
- BLAINSKI, E.; ACOSTA, E.; NOGUEIRA, P. C. P. Calibração e validação do modelo SWAT para simulação hidrológica em uma bacia hidrográfica do litoral norte catarinense. **Ambiente & Água - An Interdisciplinary Journal Of Applied Science**. Taubaté, p. 1-16. jan. 2017.
- BRAGA, B. et al. **Introdução à engenharia ambiental: o desafio do desenvolvimento sustentável**. 2 ed. São Paulo: Pearson Prentice Hall, 2005.
- BRIGHENTI, T. M. **MODELAGEM HIDROLÓGICA E AVALIAÇÃO DE DIFERENTES MÉTODOS DE CALIBRAÇÃO PARA O MODELO SWAT**. 2015. 119 f. Dissertação (Mestrado) - Curso de Programa de Pós-Graduação em Engenharia Ambiental, Universidade Federal de Santa Catarina, Florianópolis, 2015.
- CALMON, A. P. S.; SOUZA, J. C.; REIS, J. A. T. Uso combinado de curvas de permanência de qualidade e modelagem da autodepuração como ferramenta para suporte ao processo de enquadramento de cursos d'água superficiais. **Revista Brasileira de Recursos Hídricos**, Porto Alegre, v. 21, n. 1, p.1-16, 21 ago. 2015.
- CARDOSO, A. **Fundamentos da PESQUISA OPERACIONAL**. Unifal-mg: Unifal, 2011. 102 p.
- CERVO, A. L.; BERVIAN, P. A.; SILVA, R. da. **Metodologia Científica**. 6 ed. São Paulo: Pearson Prentice Hall, 2007.
- CIH. Centro Internacional de Hidroinformática, **MODELAGEM SOIL WATER ASSESSMENT TOOL (SWAT)**, 2015. Disponível em

<http://www.hidroinformatica.org/portalcih/index.php/br/>. Acesso em 02 de nov. de 2015.

DIAS, F. A. **Caracterização e análise da qualidade ambiental urbana da bacia hidrográfica do Ribeirão do Lipa**, Cuiabá/MT. Universidade Federal do Mato Grosso. Programa de Pós-Graduação em Engenharia de Edificações e Ambiental. Cuiabá – MT, fevereiro 2011. Disponível em: <http://200.129.241.80/ppgeea/sistema/dissertacoes/28.pdf>. Acesso em: 29 abril 2014.

DUTRA, W. C. P. **MODELAGEM DOS PARÂMETROS DE QUALIDADE DE ÁGUA EM TRECHO URBANIZADO DO RIO PARAIBUNA EM JUIZ DE FORA (MG)**. 2014. 56 f. Faculdade de Engenharia da UFJF, Juiz de Fora, 2014.

EMBRAPA, **Empresa Brasileira de Pesquisa Agropecuária**. Sistema brasileiro de classificação de solos. 2.ed. Rio de Janeiro: Embrapa Solos, 2006. 306p.

ESTEVES, F.A. **Fundamentos de Limnologia**. 2. ed. Rio de Janeiro: Interciência, 1998.

EYNG, E. **Controle feedforward-feedback aplicado às colunas de Absorção do processo de produção de etanol por Fermentação**. Tese de Doutorado apresentada à Faculdade de Engenharia Química para a obtenção do título de Doutor em Engenharia Química. Campinas - São Paulo, 2008.

FANTIN, L. L. D. et al. **Proposal of a methodology for pre-selection of sewage treatment systems within watersheds**. Rbrh, [s.l.], v. 22, p.1-16, 2017. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/2318-0331.011716079>.

GHORABA, S. M. **Hydrological modeling of the Simly Dam watershed (Pakistan) using GIS and SWAT model**, Alexandria University, 2015.

HAYKIN, S. **REDES NEURAIS: princípios e prática**. Hamilton, Ontário, Canadá: Artmed, 2008. 891 p. (2º).

IAPAR. **IAPAR**.2016. Disponível em: <http://www.iapar.br/>. Acesso em: 10 out. 2016.

IBGE, **Instituto brasileiro de geografia e estatística**, 2010. Disponível em <http://www.ibge.gov.br/home/estatistica/populacao/censo2010/>. Acesso em 02 nov. 2015.

IBGE, Instituto Brasileiro de Geografia e estatística. **Manual Técnico de Pedologia**. 2.ed. Rio de Janeiro, RJ. 2007. 316p.

JAIN, A.k.; MAO, J.; MOHIUDDIN, K. M. **Artificial neural networks: a tutorial**. **Computer**, [s.l.], v. 29, n. 3, p. 31-44, mar. 1996. Institute of Electrical and Electronics Engineers (IEEE). <http://dx.doi.org/10.1109/2.485891>.

KANNEL P. R.; KANEL, S. R.; LEE, S.; LEE, Y.; GAN T. Y. **A Review Of Public Domain Water Quality Models for Simulating Dissolved Oxygen in Rivers and**

Streams. Environmental Modeling & Assessment, Volume 16, 2º edição, p. 183-204, Abril de 2011.

KIESER & ASSOCIATES, **Environmental Science and Engineering, SWAT Modeling of the St. Joseph River Watershed**, Michigan and Indian, 2005.

LIMA, R. N. S. et al. **Estudo da poluição pontual e difusa na bacia de contribuição do reservatório da usina hidrelétrica de Funil utilizando modelagem espacialmente distribuída em Sistema de Informação Geográfica, Engenharia Sanitária e Ambiental**, [s.l.], v. 21, n. 1, p.139-150, mar. 2016. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s1413-41520201600100127676>.

MACHADO, R. E. **Simulação de escoamento e de produção de sedimentos em uma bacia hidrográfica utilizando técnicas de modelagem e geoprocessamento.** Piracicaba – SP. Tese (Doutorado em Agronomia) – Escola Superior de Agronomia “Luiz de Queiroz”, Universidade de São Paulo, 2002.

MACHADO, A. R. et al. **Avaliação do modelo J2000/JAMS para modelagem hidrológica em bacias hidrográficas brasileiras.** Engenharia Sanitaria e Ambiental, [s.l.], v. 22, n. 2, p.327-340, abr. 2017. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s1413-41522016145177>.

MARCHIORE, J. **AVALIAÇÃO HIDROSSEDIMENTOLÓGICA DA BACIA HIDROGRÁFICA DO RIO QUATORZE, FRANCISCO BELTRÃO (PR), POR MEIO DE MODELO HIDROLÓGICO COMPUTACIONAL.** 2014. 102 f - Curso de Engenharia Ambiental, Universidade Tecnológica Federal do Paraná, Francisco Beltrão, 2014.

MACHADO, R.E. & VETTORAZZI, C.A. **Simulação da produção de sedimentos para a microbacia hidrográfica do Ribeirão dos Marins, SP.** R. Bras. Ci. Solo, 27:735-741, 2003.

MEDIANEIRA. **Plano diretor Urbano e Rural. Áreas e localizações geográficas,** 2016. Disponível em: http://www.medianeira.pr.gov.br/planodiretor/Leis/0_Plano_Direto_de_Medianeira_c_ompleto.pdf.

MOURA, A. C. M.. **Geoprocessamento na gestão e planejamento urbano.** Belo Horizonte : Ed. Da autora, 2003, 294p

NARDINI, R.c.. **TÉCNICAS DE GEOPROCESSAMENTO PARA ANÁLISE MORFOMÉTRICA DA MICROBACIA DO RIBEIRÃO ALTO ÁGUA FRIA – BOFETE (SP).** Brazilian Journal Of Biosystems Engineering. Bofete (sp), p. 1-11. jun. 2016.

PAULA, L. M. **Avaliação da qualidade da água e autodepuração do Rio Jordão, Araguari (MG).** 2011. 196 f. Dissertação (Mestrado em Engenharia Civil) – Universidade Federal de Uberlândia, Uberlândia, 2011.

POSTGRESQL. **PostgreSQL.** 2016. Disponível em: <https://www.postgresql.org/>. Acesso em: 05 nov. 2016.

POSTGIS. PostGIS **2.3.0dev Manual**. 2016. Disponível em <http://postgis.net/docs/manual-dev/postgis-br.html>. Acesso em: 06 julho 2016.

PRADO, R. S. **Introdução Ao Arcgis Conceitos E Comandos**. Universidade Federal de Alagoas, 2006.

QUEIROZ, G. R.; FERREIRA, K. R. **SGBDs com extensões espaciais**. In: Casanova, M. A.; Câmara, G.; Davis-Jr, C. A.; Vinhas, L.; Queiroz, G. R. (Org.). Bancos de Dados Geográficos. Curitiba: Editora MundoGEO, 2005. cap. 8, p. 267-303.

SALARI, M.; SHAHID, E. S.; AFZALI, S. H.; EHTESHAMI, M.; CONTI, G. O.; DERAKHSHAN, Z.; SHEIBANI, S. N. **Quality assessment and artificial neural networks modeling for characterization of chemical and physical parameters of potable water**. Food And Chemical Toxicology, [s.l.], v. 118, p. 212-219, ago. 2018. Elsevier BV. <http://dx.doi.org/10.1016/j.fct.2018.04.036>.

SANEPAR. **Empregados recuperam mata ciliar de manancial em Medianeira**. 2016. Disponível em: <http://site.sanepar.com.br/noticias/empregados-recuperam-mata-ciliar-de-manancial-em-medianeira>. Acesso em: 02 nov. 2016.

SANTOS, I. et al. **Desenvolvimento de Modelo de Avaliação da Qualidade de Água e Transporte de Sedimentos para Pequenas e Médias Centrais Elétricas: Modelagem da Erosão e Transporte de Sedimentos**. Instituto de Tecnologia para o Desenvolvimento (LACTEC). Curitiba – PR, 2005.

SANTOS, I. **Modelagem geobiohidrológica como ferramenta no planejamento ambiental: estudo da bacia hidrográfica do rio pequeno**, São José dos Pinhais – Pr. Curitiba - PR, Dissertação (Mestrado em Agronomia) – Setor de Ciências Agrárias, Universidade Federal do Paraná, 2001.

SALLA, M. R.; PEREIRA, C. E.; ALAMY FILHO, J. E. **Estudo da autodepuração do Rio Jordão, localizado na bacia hidrográfica do Rio Dourados**. Eng Sanit Ambient, Uberlândia (mg), v. 18, n. 2, p.105-114, jun. 2013.

SALLES, L. A. **Calibração e validação do modelo SWAT para a predição de vazões na bacia do ribeirão Pipiripau**. Dissertação de mestrado, Universidade de Brasília, Brasília, DF, 2012. Disponível em: http://repositorio.unb.br/bitstream/10482/10774/1/2012_LeandroDeAlmeidaSalles.pdf. Acesso em: 15 out. 2016.

SÁ, T. F. F. et al. **SISTEMA DE INFORMAÇÕES GEOGRÁFICAS (SIG) PARA A GESTÃO AMBIENTAL DE BACIAS HIDROGRÁFICAS**. Simpósio Brasileiro de Ciências Geodésicas e Tecnologias da Geoinformação, Recife, p.1-4, 2012.

SILVA, M. M. A. P. M. et al. **Modelagem da qualidade da água na bacia hidrográfica do Rio Piracicaba (MG)**. Engenharia Sanitária e Ambiental, [s.l.], v. 22, n. 1, p.133-143, fev. 2017. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/s1413-41522017134420>.

SCHÄFER, A. G. **Integração do método da curva número em um banco de dados espacial para a estimativa da precipitação efetiva em bacias hidrográficas.** XVI Simpósio Brasileiro de Sensoriamento Remoto - Sbsr, Foz do Iguaçu, p.1-8, 13 abr. 2013.

SCHUTZ, F. C. A. **DESENVOLVIMENTO DE UM MODELO EM REDES NEURAIS PARA A SIMULAÇÃO DA AUTODEPURAÇÃO DO RIO ALEGRIA - PR.** 2014. 113 f. Tese (Doutorado) - Curso de Engenharia Agrícola, Universidade Federal de Campina Grande, Campina Grande, 2014.

SILVA, P.M.O.; MELLO, C.R.; SILVA, A.M. & COELHO, G. **Modelagem da hidrografia de cheia em uma bacia hidrográfica da região Alto Rio Grande.** R. Bras. Eng. Agric. Amb. 12:258-265, 2008.

SILVA, M. P. et al. **PLANEJAMENTO AMBIENTAL DA BACIA HIDROGRÁFICA DO CÓRREGO DA OLGA, UGRHI PONTAL DO PARANAPANEMA – SÃO PAULO.** Sociedade & Natureza,[s.l.], v. 28, n. 3, p.409-428, dez. 2016. FapUNIFESP (SciELO). <http://dx.doi.org/10.1590/1982-451320160307>.

SHEKKAR, S.; CHAWLA, S. **Spatial databases - a tour.** Upper Saddle River, NJ, USA, Prentice-Hall, 2003.

JACQUES, L. U. **ANÁLISE DAS CARACTERÍSTICAS MORFOMÉTRICAS DAS MICROBACIAS URBANAS DE MEDIANEIRA – PR, ATRAVÉS DE FERRAMENTAS DE GEOPROCESSAMENTO.** 2015. 23 f. Monografia (Especialização) - Curso de Especialização em Gestão Ambiental em Municípios, Universidade Tecnológica Federal do Paraná Diretoria de Pesquisa e Pós-graduação Especialização em Gestão Ambiental em Municípios, Medianeira, 2015.

JUNIOR, J. L. B. **Análise da disponibilidade hídrica da bacia do rio Jundiá por meio de simulações hidrológicas de cenários prováveis.** Dissertação de mestrado, Universidade Estadual de Campinas, Campinas, SP, 2014. Disponível em: <http://www.bibliotecadigital.unicamp.br/document/?code=000926942&fd=y>. Acesso em: 25 ago. 2016.

SEMA, Secretaria Do Estado E Do Meio Ambiente E Recursos Hídricos, **Bacias hidrográficas do Paraná,** 2007.

SOUZA, R. M. et al, **O modelo SWAT como ferramenta para a gestão de recursos hídricos: avaliação de vazões de outorga na bacia hidrográfica do altíssimo rio negro.** XVIII Simpósio Brasileiro de Recursos Hídricos, 2007.

TERCINI, J. R. B. **MODELAGEM DA QUALIDADE DA ÁGUA INTEGRANDO RIO E RESERVATÓRIO.** 2014. 117 f. Dissertação (Mestrado) - Curso de Engenharia Hidráulica Ambiental, Universidade de São Paulo, São Paulo, 2014.

TURRIONI, J. B.; MELLO, C. H. P. **METODOLOGIA DE PESQUISA EM ENGENHARIA DE PRODUÇÃO.** Itajubá: Unifei, 2012. 191 p.

UZEIKA, T. **Aplicabilidade Do Modelo Swat (Soil And Water Assessment Tool) Na Simulação Da Produção De Sedimentos Em Uma Pequena Bacia Hidrográfica Rural**. Porto alegre, 2009.

VERGARA, S. C. **Métodos de pesquisa em administração**. São Paulo: Atlas, 2005.

VELLASCO, M. M. B. R. **REDES NEURAIS ARTIFICIAIS**. Rio de Janeiro: Laboratório Inteligência Computacional Aplicada, 2007.

VON SPERLING, M. **Introdução à qualidade das águas e ao tratamento de esgotos**. Belo Horizonte: Departamento de Engenharia Sanitária e Ambiental, UFMG, 1996. 240 p. (Princípios do tratamento biológico de águas residuárias, v. 1).

WATERBASE. [Http://www.waterbase.org](http://www.waterbase.org). 2017. Disponível em: <http://www.waterbase.org>>. Acesso em: 01 fev. 2017.

APÊNDICE A - Banco de dados de treinamento

BANCO DE DADOS DE TREINAMENTO

dist	dist futuro	DQO	NA	T	pH	CE	Nitrito	Nitrato	vazao	OD	DQOe	DBOe	Nae	ODfuturo	DBOfuturo
0,000	141,170	8,000	0,440	27,100	7,010	78,300	0,110	0,300	0,326	6,120	97,100	6,450	90,500	4,100	7,300
141,170	280,170	8,000	0,440	27,100	7,010	78,300	0,110	0,300	0,326	6,120	97,100	6,450	90,500	4,190	7,000
280,170	350,000	8,000	0,440	27,100	7,010	78,300	0,110	0,300	0,326	6,120	97,100	6,450	90,500	7,000	3,000
350,000	500,000	8,000	0,440	27,100	7,010	78,300	0,110	0,300	0,326	6,120	97,100	6,450	90,500	8,960	2,000
500,000	500,000	8,000	0,440	27,100	7,010	78,300	0,110	0,300	0,326	6,120	97,100	6,450	90,500	8,960	2,000
0,000	141,170	9,000	0,508	25,100	6,880	77,000	0,120	0,500	0,254	6,800	90,000	13,300	77,800	4,350	7,000
141,170	350,000	9,000	0,508	25,100	6,880	77,000	0,120	0,500	0,254	6,800	90,000	13,300	77,800	7,840	2,000
350,000	500,000	9,000	0,508	25,100	6,880	77,000	0,120	0,500	0,254	6,800	90,000	13,300	77,800	8,960	1,000
500,000	500,000	9,000	0,508	25,100	6,880	77,000	0,120	0,500	0,254	6,800	90,000	13,300	77,800	8,960	1,000
0,000	141,170	8,000	0,470	26,100	6,930	81,000	0,140	0,100	0,303	7,170	106,250	4,000	110,200	4,300	7,200
141,170	153,170	8,000	0,470	26,100	6,930	81,000	0,140	0,100	0,303	7,170	106,250	4,000	110,200	2,100	11,000
153,170	500,000	8,000	0,470	26,100	6,930	81,000	0,140	0,100	0,303	7,170	106,250	4,000	110,200	8,960	2,000
500,000	500,000	8,000	0,470	26,100	6,930	81,000	0,140	0,100	0,303	7,170	106,250	4,000	110,200	8,960	2,000
0,000	141,170	7,800	0,330	26,100	7,040	76,000	0,060	0,300	0,323	7,100	112,500	49,650	119,050	4,100	7,200
141,170	280,170	7,800	0,330	26,100	7,040	76,000	0,060	0,300	0,323	7,100	112,500	49,650	119,050	4,200	5,000
280,170	350,000	7,800	0,330	26,100	7,040	76,000	0,060	0,300	0,323	7,100	112,500	49,650	119,050	7,500	3,000
350,000	500,000	7,800	0,330	26,100	7,040	76,000	0,060	0,300	0,323	7,100	112,500	49,650	119,050	8,800	2,000
500,000	500,000	7,800	0,330	26,100	7,040	76,000	0,060	0,300	0,323	7,100	112,500	49,650	119,050	8,800	2,000
0,000	141,170	7,000	0,520	25,900	7,010	74,900	0,080	0,000	0,360	8,500	240,100	97,100	224,900	5,100	7,400
141,170	153,170	7,000	0,520	25,900	7,010	74,900	0,080	0,000	0,360	8,500	240,100	97,100	224,900	3,100	12,000
153,170	280,170	7,000	0,520	25,900	7,010	74,900	0,080	0,000	0,360	8,500	240,100	97,100	224,900	5,600	4,000
280,170	350,000	7,000	0,520	25,900	7,010	74,900	0,080	0,000	0,360	8,500	240,100	97,100	224,900	9,300	2,000
350,000	500,000	7,000	0,520	25,900	7,010	74,900	0,080	0,000	0,360	8,500	240,100	97,100	224,900	8,800	2,000
500,000	500,000	7,000	0,520	25,900	7,010	74,900	0,080	0,000	0,360	8,500	240,100	97,100	224,900	8,800	2,000
0,000	141,170	8,000	2,300	22,100	6,990	111,000	0,110	0,500	0,365	6,180	213,500	105,150	223,900	4,350	7,300
141,170	153,170	8,000	2,300	22,100	6,990	111,000	0,110	0,500	0,365	6,180	213,500	105,150	223,900	3,400	13,000
153,170	280,170	8,000	2,300	22,100	6,990	111,000	0,110	0,500	0,365	6,180	213,500	105,150	223,900	6,600	3,000

280,170	350,000	8,000	2,300	22,100	6,990	111,000	0,110	0,500	0,365	6,180	213,500	105,150	223,900	7,840	2,000
350,000	500,000	8,000	2,300	22,100	6,990	111,000	0,110	0,500	0,365	6,180	213,500	105,150	223,900	8,400	2,000
500,000	500,000	8,000	2,300	22,100	6,990	111,000	0,110	0,500	0,365	6,180	213,500	105,150	223,900	8,400	2,000
0,000	141,170	2,000	0,710	17,500	6,890	99,000	0,080	0,000	0,373	7,800	124,000	35,950	175,200	4,000	7,200
141,170	153,170	2,000	0,710	17,500	6,890	99,000	0,080	0,000	0,373	7,800	124,000	35,950	175,200	3,100	10,000
153,170	280,170	2,000	0,710	17,500	6,890	99,000	0,080	0,000	0,373	7,800	124,000	35,950	175,200	4,600	6,400
280,170	500,000	2,000	0,710	17,500	6,890	99,000	0,080	0,000	0,373	7,800	124,000	35,950	175,200	8,200	3,900
500,000	500,000	2,000	0,710	17,500	6,890	99,000	0,080	0,000	0,373	7,800	124,000	35,950	175,200	8,200	3,900
0,000	141,170	4,100	0,750	22,000	7,120	81,300	0,020	0,050	0,332	7,180	94,430	10,875	294,100	3,000	10,000
141,170	153,170	4,100	0,750	22,000	7,120	81,300	0,020	0,050	0,332	7,180	94,430	10,875	294,100	2,850	11,000
153,170	280,170	4,100	0,750	22,000	7,120	81,300	0,020	0,050	0,332	7,180	94,430	10,875	294,100	5,600	5,300
280,170	350,000	4,100	0,750	22,000	7,120	81,300	0,020	0,050	0,332	7,180	94,430	10,875	294,100	7,320	3,200
350,000	500,000	4,100	0,750	22,000	7,120	81,300	0,020	0,050	0,332	7,180	94,430	10,875	294,100	8,030	4,500
500,000	500,000	4,100	0,750	22,000	7,120	81,300	0,020	0,050	0,332	7,180	94,430	10,875	294,100	8,030	4,500
0,000	141,170	4,200	0,400	21,200	7,110	90,000	0,120	0,100	0,362	7,480	114,800	21,500	293,000	3,230	10,200
141,170	153,170	4,200	0,400	21,200	7,110	90,000	0,120	0,100	0,362	7,480	114,800	21,500	293,000	3,920	10,600
153,170	280,170	4,200	0,400	21,200	7,110	90,000	0,120	0,100	0,362	7,480	114,800	21,500	293,000	4,430	7,700
280,170	350,000	4,200	0,400	21,200	7,110	90,000	0,120	0,100	0,362	7,480	114,800	21,500	293,000	7,140	3,200
350,000	500,000	4,200	0,400	21,200	7,110	90,000	0,120	0,100	0,362	7,480	114,800	21,500	293,000	5,940	3,600
500,000	500,000	4,200	0,400	21,200	7,110	90,000	0,120	0,100	0,362	7,480	114,800	21,500	293,000	5,940	3,600
0,000	141,170	2,000	0,460	22,900	5,990	72,000	0,090	0,300	0,354	7,700	165,000	40,250	40,000	4,380	7,200
141,170	280,170	2,000	0,460	22,900	5,990	72,000	0,090	0,300	0,354	7,700	165,000	40,250	40,000	6,320	2,000
280,170	350,000	2,000	0,460	22,900	5,990	72,000	0,090	0,300	0,354	7,700	165,000	40,250	40,000	7,400	3,000
350,000	500,000	2,000	0,460	22,900	5,990	72,000	0,090	0,300	0,354	7,700	165,000	40,250	40,000	7,800	2,000
500,000	500,000	2,000	0,460	22,900	5,990	72,000	0,090	0,300	0,354	7,700	165,000	40,250	40,000	7,800	2,000
0,000	141,170	2,000	0,770	22,100	5,950	63,300	0,110	0,100	0,355	7,710	75,980	20,100	54,000	4,160	7,400
141,170	153,170	2,000	0,770	22,100	5,950	63,300	0,110	0,100	0,355	7,710	75,980	20,100	54,000	3,360	10,300
153,170	280,170	2,000	0,770	22,100	5,950	63,300	0,110	0,100	0,355	7,710	75,980	20,100	54,000	4,000	4,900
280,170	350,000	2,000	0,770	22,100	5,950	63,300	0,110	0,100	0,355	7,710	75,980	20,100	54,000	6,290	4,100
350,000	500,000	2,000	0,770	22,100	5,950	63,300	0,110	0,100	0,355	7,710	75,980	20,100	54,000	7,340	3,100
500,000	500,000	2,000	0,770	22,100	5,950	63,300	0,110	0,100	0,355	7,710	75,980	20,100	54,000	7,340	3,100
0,000	141,170	7,800	0,930	19,300	7,000	68,500	0,280	0,240	0,319	7,000	86,500	8,900	234,000	4,100	7,000

141,170	280,170	7,800	0,930	19,300	7,000	68,500	0,280	0,240	0,319	7,000	86,500	8,900	234,000	5,200	6,000
280,170	350,000	7,800	0,930	19,300	7,000	68,500	0,280	0,240	0,319	7,000	86,500	8,900	234,000	7,500	3,000
350,000	500,000	7,800	0,930	19,300	7,000	68,500	0,280	0,240	0,319	7,000	86,500	8,900	234,000	8,000	2,000
500,000	500,000	7,800	0,930	19,300	7,000	68,500	0,280	0,240	0,319	7,000	86,500	8,900	234,000	8,000	2,000
0,000	141,170	7,200	1,000	18,000	6,000	68,500	0,230	0,120	0,348	7,200	86,520	8,900	233,000	4,960	7,500
141,170	153,170	7,200	1,000	18,000	6,000	68,500	0,230	0,120	0,348	7,200	86,520	8,900	233,000	4,800	9,200
153,170	280,170	7,200	1,000	18,000	6,000	68,500	0,230	0,120	0,348	7,200	86,520	8,900	233,000	5,750	6,400
280,170	350,000	7,200	1,000	18,000	6,000	68,500	0,230	0,120	0,348	7,200	86,520	8,900	233,000	6,230	3,400
350,000	500,000	7,200	1,000	18,000	6,000	68,500	0,230	0,120	0,348	7,200	86,520	8,900	233,000	7,200	1,600
500,000	500,000	7,200	1,000	18,000	6,000	68,500	0,230	0,120	0,348	7,200	86,520	8,900	233,000	7,200	1,600
0,000	141,170	2,000	0,580	23,100	6,830	83,000	0,010	0,030	0,335	7,300	94,050	33,750	212,000	5,000	6,900
141,170	153,170	2,000	0,580	23,100	6,830	83,000	0,010	0,030	0,335	7,300	94,050	33,750	212,000	2,120	11,200
153,170	280,170	2,000	0,580	23,100	6,830	83,000	0,010	0,030	0,335	7,300	94,050	33,750	212,000	5,000	6,000
280,170	350,000	2,000	0,580	23,100	6,830	83,000	0,010	0,030	0,335	7,300	94,050	33,750	212,000	7,000	3,000
350,000	350,000	2,000	0,580	23,100	6,830	83,000	0,010	0,030	0,335	7,300	94,050	33,750	212,000	7,000	3,000

APÊNDICE B- Algoritmo RNA1

ALGORITMO RNA1

```
#!/usr/bin/env python
# coding: utf-8
# Regressão: Demanda de OD
import pathlib
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from keras.wrappers.scikit_learn import KerasRegressor
from sklearn.model_selection import cross_val_score
from keras.models import Sequential
from keras.layers import Dense

print(tf.__version__)

#Utilizando o pandas, importe os dados:

column_names = ['DBOfuturo','DQO', 'NA', 'T', 'pH', 'CE', 'Nitrito','Nitrato',
'vazao','Nae', 'OD']
raw_dataset = base = pd.read_csv('Banco de dados de treinamento final.csv',
error_bad_lines=False)
dataset = raw_dataset.copy()
dataset.tail()

### Limpando os dados

### se a base contém alguns valores não conhecidos (*unknown*).
```

```
dataset.isna().sum()

#remova as linhas com esses valores não conhecidos.

dataset = dataset.dropna()

dataset.tail()

### Separando dados de treinamento e teste

train_dataset = dataset.sample(frac=0.8,random_state=0)
test_dataset = dataset.drop(train_dataset.index)

### Inspeccionado os dados

### Como está a distribuição de algumas colunas do conjunto de treinamento.

sns.pairplot(train_dataset[['DBOfuturo','DQO', 'NA', 'T', 'pH', 'CE', 'Nitrito','Nitrato',
'vazao','Nae', 'OD']], diag_kind="kde")

train_stats = train_dataset.describe()
train_stats.pop("OD")
train_stats = train_stats.transpose()
train_stats

### Separe features de labels

train_labels = train_dataset.pop('OD')
test_labels = test_dataset.pop('OD')

### Normalize os dados

def norm(x):
    return (x - train_stats['mean']) / train_stats['std']
```

```
normed_train_data = norm(train_dataset)
normed_test_data = norm(test_dataset)
```

O dado normalizado é o que nós usaremos para treinar o modelo.

A estatística usada para normalizar as entradas aqui (média e desvio padrão)

O Modelo

Construindo o modelo

```
def build_model():
    model = keras.Sequential([
        layers.Dense(10, input_shape=[len(train_dataset.keys())]),
        layers.Dense(10),
        layers.Dense(1)
    ])
```

```
optimizer = tf.keras.optimizers.RMSprop(0.001)
```

```
model.compile(loss='mse',
              optimizer=optimizer,
              metrics=['mae', 'mse'])
```

```
return model
```

```
model = build_model()
```

```
model.summary()
```

```
example_batch = normed_train_data[:10]
example_result = model.predict(example_batch).flatten()
example_result
```

Treinando o modelo

```
# Mostra o progresso do treinamento imprimindo um único ponto para cada
epoch completada
```

```
class PrintDot(keras.callbacks.Callback):
    def on_epoch_end(self, epoch, logs):
        if epoch % 100 == 0: print("")
        print('.', end="")
```

```
EPOCHS = 1000
```

```
history = model.fit(
    normed_train_data, train_labels, epochs=EPOCHS, validation_split = 0.2,
    verbose=0, callbacks=[PrintDot()])
```

```
# armazenados no objeto `history` o progresso
```

```
hist = pd.DataFrame(history.history)
hist['epoch'] = history.epoch
hist.tail()
```

```
def plot_history(history):
    hist = pd.DataFrame(history.history)
    hist['epoch'] = history.epoch
```

```
plt.figure()
plt.xlabel('Epoch')
plt.ylabel('Mean Abs Error [MPG]')
plt.plot(hist['epoch'], hist['mae'],
         label='Train Error')
plt.plot(hist['epoch'], hist['val_mae'],
         label = 'Val Error')
plt.ylim([0,5])
plt.legend()
```

```
plt.figure()
plt.xlabel('Epoch')
plt.ylabel('Mean Square Error [MPG^2]')
plt.plot(hist['epoch'], hist['mse'],
         label='Train Error')
plt.plot(hist['epoch'], hist['val_mse'],
         label = 'Val Error')
plt.ylim([0,20])
plt.legend()
plt.show()

plot_history(history)

model = build_model()

# O parâmetro patience é o quantidade de epochs para checar as melhoras
early_stop = keras.callbacks.EarlyStopping(monitor='val_loss', patience=20)

history = model.fit(normed_train_data, train_labels, epochs=EPOCHS,
                    validation_split = 0.2, verbose=0, callbacks=[early_stop, PrintDot()])

plot_history(history)

loss, mae, mse = model.evaluate(normed_test_data, test_labels, verbose=2)

print("Testing set Mean Abs Error: {:.2f} OD".format(mae))

### predições

test_predictions = model.predict(normed_test_data).flatten()

plt.scatter(test_labels, test_predictions, color='red')
plt.xlabel('True Values [OD]')
plt.ylabel('Predictions [OD]')
```



```

plt.axis('equal')
plt.axis('square')
plt.xlim([0,plt.xlim()[1]])
plt.ylim([0,plt.ylim()[1]])
_ = plt.plot([-100, 100], [-100, 100])

test_predictions

test_labels

def mean_absolute_percentage_error(y_true, y_pred):
    y_true, y_pred = np.array(y_true), np.array(y_pred)
    return np.mean(np.abs((y_true - y_pred) / y_true)) * 100

from sklearn.metrics import mean_squared_error, mean_absolute_error

MAE = mean_absolute_error(test_labels,test_predictions)
MSE = mean_squared_error(test_labels,test_predictions)
RMSE = np.sqrt(MSE)
MAPE = mean_absolute_percentage_error(test_labels,test_predictions)

print("MAE = {:.10f}".format(MAE))
print("MSE = {:.10f}".format(MSE))
print("RMSE = {:.10f}".format(RMSE))
print("MAPE = {:.10f}%".format(MAPE))

error = test_predictions - test_labels
plt.hist(error, bins = 25)
plt.xlabel("Prediction Error [OD]")
_ = plt.ylabel("Count")

print(model.get_weights())

```

```
regressor = KerasRegressor(build_fn = build_model, batch_size = 14, epochs =
1000)
# validação cruzada

accuracies = cross_val_score(estimator = regressor, X = normed_train_data, y
= train_labels,scoring='r2',cv = 3, n_jobs = 1)

mean = accuracies.mean()
variance = accuracies.std()

print(accuracies)

mean

variance
]
```