



**UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE RUSSAS
GRADUAÇÃO EM ENGENHARIA MECÂNICA**

JOÃO PAULO ALVES DOS SANTOS NUNES

**APLICAÇÃO DE REDES NEURAIIS PARA A PREVISÃO DE SÉRIES TEMPORAIS,
REFERENTES AO CONSUMO DE GÁS NITROGÊNIO EM UMA EMPRESA DE
MARGARINA DO ESTADO DO CEARÁ**

RUSSAS

2022

JOÃO PAULO ALVES DOS SANTOS NUNES

APLICAÇÃO DE REDES NEURAIS PARA A PREVISÃO DE SÉRIES TEMPORAIS,
REFERENTES AO CONSUMO DE GÁS NITROGÊNIO EM UMA EMPRESA DE
MARGARINA DO ESTADO DO CEARÁ

Trabalho de Conclusão de Curso apresentado
ao curso de Engenharia Mecânica da
Universidade Federal do Ceará, Campus
Russas, como requisito parcial para obtenção
do título de Bacharel em Engenharia Mecânica.

Orientador: Profa. Dra. Caroliny Gomes de
Oliveira

RUSSAS

2022

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Biblioteca Universitária

Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

C247a Nunes, João Paulo Alves dos Santos.

Aplicação de redes neurais para a previsão de séries temporais, referentes ao consumo de gás nitrogênio em uma empresa de margarina do estado do ceará/

João Paulo Alves dos
Santos Nunes. – 2022.
74 f.: il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas, Curso de Engenharia Mecânica, Russas, 2022. Orientação: Profa. Dra. Caroliny Gomes de Oliveira.

1. Redes Neurais. 2. Indústria 4.0. 3. Previsão. I. Título.

CDD 620.1

JOÃO PAULO ALVES DOS SANTOS NUNES

APLICAÇÃO DE REDES NEURAIIS PARA A PREVISÃO DE SÉRIES TEMPORAIS,
REFERENTES AO CONSUMO DE GÁS NITROGÊNIO EM UMA EMPRESA DE
MARGARINA DO ESTADO DO CEARÁ

Trabalho de Conclusão de Curso apresentado
ao curso de Engenharia Mecânica da
Universidade Federal do Ceará, Campus
Russas, como requisito parcial para obtenção
do título de Bacharel em Engenharia Mecânica.

Aprovado em: __ / __ / ____.

BANCA EXAMINADORA

Profa. Dra. Caroliny Gomes de Oliveira (Orientadora)
Universidade Federal do Ceará (UFC)

Prof. Dr. Bonfim Amaro Júnior
Universidade Federal do Ceará (UFC)

Prof. Dr. Markos de Oliveira Freitas
Universidade Federal do Ceará (UFC)

AGRADECIMENTOS

Gostaria primeiramente de agradecer a Deus por me abençoar todos os dias desde que eu nasci.

Também gostaria de agradecer a meu pai, Raimundo Nonato Nunes, por todos os ensinamentos e conselhos sempre que possível; a minha mãe, Rita de Cacia Alves dos Santos Nunes, pelo apoio incondicional desde o berço; a meu irmão, Pedro Paulo Alves dos Santos Nunes, por sempre me ajudar nos “perrengues” pessoais e profissionais da vida e a toda minha família, em especial a família Bica, pelo suporte e pelo amor, melhor família não há e minha gratidão por todos será eterna.

Aos meus companheiros de caminhada dentro da universidade, que fizeram os dias mais leves e a rotina menos pesada, que tenho orgulho em chamar de amigos, saibam que estarei aqui para tudo o que precisarem, são eles: Minha namorada Maria Eduarda Gonçalves Fontes, Pedro Henrique Macena, Gilvan Antônio Cappi, João Victor Nogueira (Pias); Bruno Herculano (Pomps); Lardner Dias (Piras); Orismildes Neto e a todos que também estiveram comigo durante esse percurso.

Um agradecimento especial a Ismael Francisco e a Darlan Leocádio, por abrirem as portas da empresa para minha coleta de dados, a toda a equipe de administração que se colocou à disposição para responder minhas perguntas, sem vocês, esse trabalho não seria possível.

A Profa. Dra. Caroliny Gomes de Oliveira, pela excelente orientação e por estar sempre a disposição para tirar as minhas dúvidas.

Aos professores participantes da banca examinadora: Prof. Dr. Bonfim Amaro Júnior e Prof. Dr. Markos de Oliveira Freitas pelo tempo, pelas valiosas colaborações e sugestões.

A Universidade Federal do Ceará - Campus Russas, por todo o conhecimento transmitido ao longo desses anos, bem como todas as oportunidades de desenvolvimento de novos projetos e ideias que me fizeram pensar fora da caixa e ir buscar novos meios de transformar o ambiente ao meu redor, continue criando sonhadores e pessoas criativas.

“O insucesso é apenas uma oportunidade para recomeçar com mais inteligência.” (Henry Ford).

RESUMO

Em um mundo que se encontra vivendo sua quarta revolução industrial, ou indústria 4.0, o uso dos processos de digitalização se tornou fundamental para a sobrevivência de qualquer empresa, independentemente de seu setor de atuação, tendo em vista que essas operações permitem não apenas um maior conhecimento da empresa, para com os seus próprios procedimentos e variáveis, mas fornecem uma vantagem comercial em relação as suas concorrentes. Diante deste novo tipo de desafio, empresas vem buscando maneiras de realizar acompanhamentos e previsões de suas principais variáveis de processos, dentre diversas outras abordagens. Portanto, o trabalho em questão possui uma abordagem prática acerca de uma aplicação viável, para a previsão do valor consumido de gás nitrogênio (N_2), levando-se em consideração todos os processos requisitantes deste tipo de material, em uma empresa produtora de margarina, situada no estado do Ceará. Para a realização de tal atividade, foram utilizadas ferramentas denominadas redes neurais, bem como a análise e tratamento de dados reais de operação da empresa, que foram coletados digitalmente através do auxílio de hardwares específicos para tal, que forneceram valores no tempo, referentes às condições de utilização do N_2 . Pode-se constatar que os modelos aplicados atuaram de forma satisfatória com o proposto, obtendo valores futuros dos aspectos comportamentais para as condições de utilização do produto em estudos, permitindo à empresa detentora dos dados, um maior controle na aplicação deste produto.

Palavras-chave: Industria 4.0; Redes Neurais; Gás Nitrogênio; Estatística.

ABSTRACT

In a world that is living in its fourth industrial revolution, or industry 4.0, the use of digitization processes have become fundamental to the survival of any company, regardless of its sector of operation, in view of the fact that these operations allow not only a greater knowledge of the company, with its own procedures and variables, but provide a commercial advantage over its competitors. Faced with this new type of challenge, companies have been looking for ways to follow up and forecast their main process variables, among several other approaches. Therefore, the work in question has a practical approach about a viable application for the prediction of the consumed value of nitrogen gas (N₂), taking into account all the requisite processes of this type of material, in a margarine producing company, located in the state of Ceará. To carry out this activity, tools called neural networks were used, as well as the analysis and processing of real data of the company's operation, which were collected digitally through the aid of specific hardware, which provided values in time, conditions of use of N₂. It can be seen that the models applied acted satisfactorily with the proposed, obtaining future values of behavioral aspects for the conditions of use of the product in studies, allowing the company holding the data, a greater control in the application of this product.

Keywords: Industry 4.0; Neural Networks; Nitrogen gas; Statistics.

LISTA DE FIGURAS

Figura 1 – Revoluções Industriais.....	18
Figura 2 – Pilares da Indústria 4.0.....	20
Figura 3 – Computação em Nuvem.....	21
Figura 4 – Conceito do Big Data.....	23
Figura 5 – Relação entre IA, Machine learning e Deep learning.....	26
Figura 6 – Ciclo de implementação manual do filtro de spam.....	27
Figura 7 – Utilizando um algoritmo de Machine Learning para classificar os e-mails..	28
Figura 8 – Diferença de Machine Learning e Deep Learning.....	29
Figura 9 – Constituição de uma célula neural.....	30
Figura 10 – Esquema de unidade McCulloch - Pitts.....	32
Figura 11 – Funções de Ativação.....	32
Figura 12 – Arquitetura de uma rede neural.....	33
Figura 13 – Quadro geral dos tipos de redes neurais.....	36
Figura 14 – Diagrama do Perceptron e Feed Forward.....	37
Figura 15 – Diagrama da RNN.....	38
Figura 16 – Representação do processo de recorrência nos nós das camadas ocultas....	39
Figura 17 – Diagrama da AE.....	40
Figura 18 – Série Temporal.....	41
Figura 19 – Linha de Produção da Empresa.....	47
Figura 20 – Caminho do gás nitrogênio na linha de produção.....	48
Figura 21 – Hardwares utilizados para captação dos dados dos equipamentos.....	49
Figura 22 – Ilustração da alocação dos hardwares.....	49
Figura 23 – Fluxo dos dados.....	50
Figura 24 – Representação Ilustrativa de um código de extração.....	53
Figura 25 – Linhas de código do consumo de gás na refinaria.....	53
Figura 26 – Linhas de código do consumo de gás na refinaria, agrupado.....	54
Figura 27 – Código de extração completo.....	56
Figura 28 – Arquivos Extraídos.....	57
Figura 29 – Bibliotecas Importadas.....	59
Figura 30 – Criação do Data Frame.....	60
Figura 31 – Data Frame Geral.....	60
Figura 32 – Renomeando os Nomes das Variáveis.....	61
Figura 33 – Data Frame Renomeado.....	61
Figura 34 – Plotagem dos dados.....	62
Figura 35 – Consumo de Gás, em m ³	62
Figura 36 – Pressão de Gás, em bar.....	63
Figura 37 – Temperatura de Gás, em °C.....	63
Figura 38 – Função do Coeficiente de Correlação de Spearman.....	64
Figura 39 – Matriz de correlação das variáveis.....	64
Figura 40 – Códigos de histogramas e box plots.....	66
Figura 41 – Histograma de Consumo de Gás.....	67
Figura 42 – Box Plot de Consumo de Gás.....	67
Figura 43 – Histograma de Pressão de Gás.....	68
Figura 44 – Box Plot de Pressão de Gás.....	68
Figura 45 – Histograma de Temperatura de Gás.....	69

Figura 46 – Box Plot de Temperatura de Gás.....	69
Figura 47 – Descrição de Box Plots.....	70
Figura 48 – Representação esquemática da função loop.....	71
Figura 49 – Fórmulas de Tratamento de Outliers Aplicadas em Funções Loop.....	71
Figura 50 – Histograma de Consumo de Gás Tratado.....	72
Figura 51 – Box Plot de Consumo de Gás Tratado.....	72
Figura 52 – Histograma de Pressão de Gás Tratado.....	73
Figura 53 – Box Plot de Pressão de Gás Tratado.....	73
Figura 54 – Histograma de Temperatura de Gás Tratado.....	74
Figura 55 – Box Plot de Temperatura de Gás Tratado.....	74
Figura 56 – Função de resetar indicie.....	75
Figura 57 – Representação Gráfica do Dimensionamento dos Dados.....	76
Figura 58 – Função de Normalização dos Dados.....	76
Figura 59 – Divisão do Conjunto de Dados.....	77
Figura 60 – Separação das variáveis a serem preditas.....	78
Figura 61 – Representação do <i>Dataset</i> de Treinamento.....	78
Figura 62 – Função de Preparação dos Dados.....	79
Figura 63 – Criação das amostras de treino e teste.....	79
Figura 64 – Representação das camadas da RNN.....	80
Figura 65 – Ilustração das conexões existentes entre os nós de uma rede neural.....	81
Figura 66 – Ilustração das operações realizadas nas camadas recorrentes.....	81
Figura 67 – Diagrama do Modelo da RNN Utilizada.....	82
Figura 68 – Código da Estrutura da Rede Neural.....	83
Figura 69 – Definição das medidas avaliativas.....	84
Figura 70 – Definição das configurações da RN.....	85
Figura 71 – Épocas dos processos.....	85
Figura 72 – Representação Gráfica do R^2	86
Figura 73 – Categorias de R^2	87
Figura 74 – Criação dos <i>Datasets</i> de Predição, e Função R^2 para cada Variável.....	88
Figura 75 – Modelo aplicado aos dados de teste de consumo.....	89
Figura 76 – Modelo aplicado aos dados de teste de pressão.....	89
Figura 77 – Modelo aplicado aos dados de teste de temperatura.....	89
Figura 78 – Função para mostrar o resumo geral do modelo.....	90
Figura 79 – Plotagem dos modelos aplicados ao período estudado.....	91
Figura 80 – Criação da coluna de valores preditos.....	92
Figura 81 – Data frame de Consumo, com valores preditos.....	92
Figura 82 – Utilizar a última sequência de dados para prever valores futuros.....	94
Figura 83 – Plotagem do modelo com duas horas futuras.....	95
Figura 84 – Valores dos R^2	96
Figura 85 – Resumo do modelo de previsão de Consumo de Gás Nitrogênio.....	97
Figura 86 – Resumo do modelo de previsão da Pressão de Gás Nitrogênio.....	99
Figura 87 – Resumo do modelo de previsão da Temperatura de Gás Nitrogênio.....	100
Figura 88 – Desempenho do modelo de consumo, com os dados dos sensores.....	101
Figura 89 – Desempenho do modelo de pressão, com os dados dos sensores.....	102
Figura 90 – Desempenho do modelo de temperatura, com os dados dos sensores.....	102
Figura 91 – Previsão de 2 horas futuras, para o Consumo em escala completa.....	103

Figura 92 – Previsão de 2 horas futuras, para o Consumo em escala reduzida.....	103
Figura 93 – Previsão de 2 horas futuras, para a Pressão em escala completa.....	104
Figura 94 – Previsão de 2 horas futuras, para a Pressão em escala reduzida.....	105
Figura 95 – Previsão de 2 horas futuras, para a Temperatura em escala completa.....	105
Figura 96 – Previsão de 2 horas futuras, para a Temperatura em escala reduzida.....	106

LISTA DE TABELAS

Tabela 01 – Exemplo de Tabela de Dados Brutos.....	51
Tabela 02 – Exemplo de Tabela de Dados Tratados.....	51
Tabela 03 – Cadastro de tags.....	52
Tabela 04 – Tabela Pré-extraída do banco de dados.....	54
Tabela 05 – Tabela tratada e extraída do banco de dados.....	55
Tabela 06 – Tabela de Dados Completa.....	57

LISTA DE EQUAÇÕES

Equação 01 – Estado oculto dos nós, no tempo.....	39
Equação 02 – Equação da regressão linear simples.....	43
Equação 03 – Equação da regressão linear múltipla.....	44
Equação 04 – Limites superior e inferior em um box plot.....	70
Equação 05 – Formula do escalonamento de um conjunto de dados.....	75
Equação 06 – Equação do R^2	87
Equação 07 – Equação do Erro Quadrático Médio.....	90

SUMÁRIO

1	INTRODUÇÃO.....	16
1.1	Objetivo geral	17
1.2	Objetivo geral	17
2	REFERENCIAL TEÓRICO.....	18
2.1	A quarta revolução industrial	18
2.2	Os pilares da indústria 4.0	20
2.2.1	Computação em Nuvem	20
2.2.2	Big Data.....	22
2.2.3	Internet das coisas (Internet of Things – IoT)	23
2.2.4	Digitalização.....	24
2.3	Inteligência Artificial, machine learning e deep learning.....	24
2.3.1	Inteligência Artificial.....	24
2.3.2	Machine Learning.....	25
2.3.3	Deep Learning	28
2.4	Redes Neurais	29
2.5	Tipos de Redes Neurais	35
2.5.1	Perceptron (P) e Feed Foward (FFN)	37
2.5.2	Recurrent Neural Network (RNN)	37
2.5.3	Auto-encoder (AE)	40
2.6	Séries Temporais	41
2.6.1	Autocorrelação.....	42
2.6.2	Tendência.....	42
2.6.3	Sazonalidade	42
2.6.4	Estacionariedade	42
2.6.5	Regressão Linear	43
3	METODOLOGIA.....	45
3.1	Variáveis de processo	46
3.2	Coleta de Dados.....	48
3.3	Tratamento e Extração dos Dados em SQL.....	53
3.4	Análise dos dados via Python.....	58
3.4.1	Bibliotecas Utilizadas	58
3.4.2	Criação de Data Frames.....	59
3.4.3	Análise Gráfica	61
3.4.4	Análises Estatísticas	64
3.4.5	Divisão das Amostras	75
3.4.6	Criação do Modelo	80
3.4.7	Medidas Avaliativas	86
3.4.8	Análise Preditiva.....	90

3.4.9	Previsão dos Parâmetros	93
4	RESULTADOS E DISCUSSÃO.....	96
4.1	Precisão do modelo.....	96
4.2	Valores Preditos.....	100
4.3	Previsões das variáveis	102
5	CONSIDERAÇÕES FINAIS	107
	REFERÊNCIAS	109

1 INTRODUÇÃO

A busca por excelência por partes das empresas se dá com o aumento da competitividade no atual cenário de indústria 4.0, com a premissa sempre de fazer a produção crescer, os custos decrescerem e a qualidade requerida se manter a um grau que visa atender os parâmetros de qualidade do mercado em que a mesma está inserida, bem como a constante automatização dos processos industriais.

Em um cenário como esse, manter a competitividade a um estágio crescente se torna um desafio. A indústria de margarinas abordada no trabalho, encontra-se nesse patamar, por isso, o setor busca constantemente otimizar seus processos ao máximo, seja inovando seus produtos, seja tentando diminuir seus custos de produção e buscando aumentar a eficiência dos processos. Dentre as estratégias usadas, o setor de automação e análise de dados acabam sendo o foco destas mudanças.

A pesquisa foi desenvolvida por meio de uma visita técnica em uma empresa do segmento alimentício, produtora de margarinas, gorduras vegetais e especiarias, do estado do Ceará. Foi notado que o uso de tecnologias que já estavam sendo utilizadas para o monitoramento dos processos produtivos da empresa, poderia vir a serem utilizados para a obtenção de análises mais sofisticadas, focadas na previsão de valores e na identificação de padrões específicos dos processos, o que permitiriam a companhia possuir um maior controle de toda a sua camada produtiva. Tudo isso com o auxílio de linguagens computacionais, embasadas nas teorias que cercam o assunto de redes neurais. As redes neurais foram escolhidas para a realização deste trabalho por conta de sua versatilidade e capacidade de aprendizado para diferentes tipos de problemas.

Durante a avaliação da empresa, elencou-se possíveis setores da companhia que representassem impacto considerável no desempenho de sua linha produtiva. Dentre os pontos levados em consideração, foi escolhido o manuseio de Gás Nitrogênio da companhia, pois o mesmo já possuía o aparato tecnológico necessário para a realização do estudo, diferentemente de outros setores, além de estar presente em grande parte dos procedimentos de fabricação de margarina.

Conhecer os valores futuros de desempenho do N₂ pode contribuir com a companhia a obter um maior controle de sua linha produtiva, devido ao fato de que a mesma poderá antecipar, com uma certa precisão, o comportamento e os gastos atrelados a utilização de gás.

Além de que, os dados fornecidos poderão serem aplicados em ferramentas de controle e automação de processos, proporcionando a empresa uma redução considerável dos erros associados a mão de obra humana e alterações nas qualidades do produto final.

Um desafio enfrentado pela empresa, que instigou ainda mais a realização do projeto, foi a necessidade de que os valores de temperatura e pressão do gás, em contato com o óleo a ser transformado em margarina, devem se manter com o mínimo de variação possível, pois isto poderia provocar uma troca de energia entre os meios, ocasionando em mudanças nas propriedades do óleo, o que poderia invalidar sua utilização, gerando gastos desnecessários.

1.1 Objetivo geral

O presente trabalho tem como objetivo realizar uma previsão das principais variáveis, referentes ao consumo de gás nitrogênio no setor de Refinaria, em uma empresa produtora de margarinas, utilizando-se modelos de redes neurais artificiais.

1.2 Objetivo geral

Neste trabalho, pretende-se de avaliar se os modelos se adequaram corretamente aos dados coletados por meio dos equipamentos de medição da empresa, tal como utilizar estes modelos para estimar quais serão os valores das variáveis, para um determinado tempo futuro.

Espera-se que os modelos obtidos possam vir a serem utilizados, pela produtora de margarinas, como dados relevantes para as tomadas de decisões administrativas da companhia, pois os modelos poderão estimar o possível consumo de gás para um período posterior, o que os dará uma previsibilidade melhor dos gastos futuros.

Além da utilização citada anteriormente, espera-se que o modelo possa vir a ser adequado para ser aplicado em métodos de controle e automação na empresa, ajudando a controlar a faixa de operação de temperatura e pressão do gás.

Para que os modelos de redes neurais possam ser classificados como aceitáveis para quaisquer aplicações, os mesmos deverão possuir um erro consideravelmente baixo em comparação aos dados reais, para tanto, modelos cujo os valores de R-quadrado sejam maiores que 60% e com erros quadráticos médios próximos de 0, serão considerados como valores aceitáveis para o presente trabalho.

2 REFERENCIAL TEÓRICO

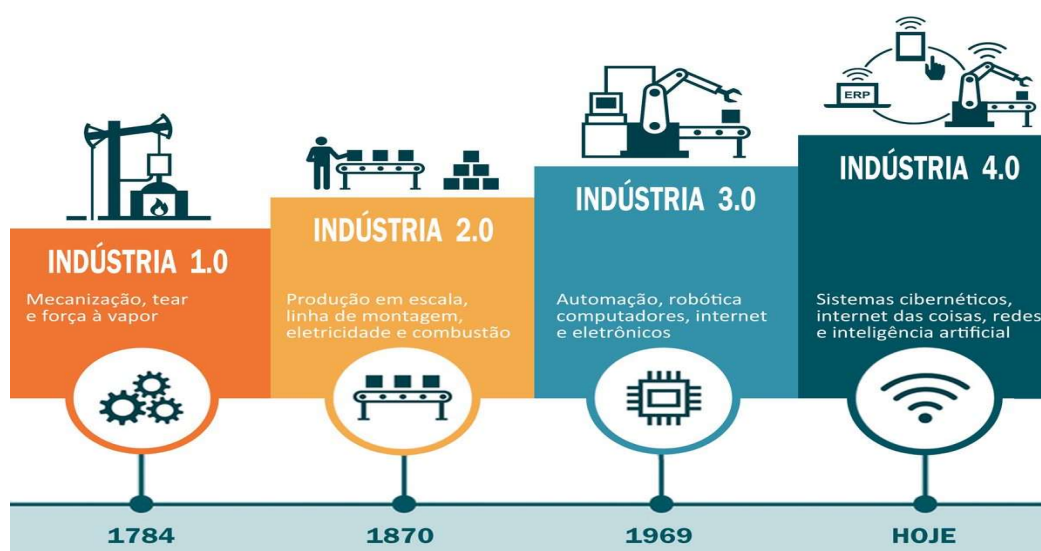
Nesta seção, estão descritos os principais pontos, referentes a utilização de redes neurais e estudos voltados para a aplicação dos conceitos da quarta revolução industrial.

2.1 A quarta revolução industrial

Inteligência artificial, robótica, nuvem e internet das coisas. Termos que há alguns anos não eram nada conhecidos, hoje já fazem parte do cotidiano de todos nós. São tecnologias que fazem parte de um conceito bem familiar no setor industrial, a Indústria 4.0. Batizada também de “4ª Revolução Industrial”, esse fenômeno está mudando, em grande escala, a automação e troca de dados, bem como as etapas de produção e os modelos de negócios, por meio do uso de máquinas e computadores.

A expressão Indústria 4.0 ou Quarta Revolução Industrial surgiu na Alemanha, de um projeto estratégico de alta tecnologia do governo. A Primeira Revolução Industrial ocorreu entre 1760 e 1840 e teve como foco a mecanização da produção através da energia do vapor. A Segunda Revolução Industrial (1870 a 1945) se baseou na energia elétrica para promover a automação da produção em massa. A Terceira Revolução Industrial (1950 a 1970) digitalizou a produção através de aparelhos, dispositivos eletrônicos e da tecnologia da informação. A Quarta Revolução Industrial busca a informatização da manufatura, e foi mencionada pela primeira vez na Hannover Messe em outubro de 2012. Na Figura 1, temos uma representação das revoluções industriais.

Figura 1 – Revoluções Industriais.



Fonte: Adaptado de Wert Ambiental (2022)

A Indústria 4.0 tem impacto significativo na produtividade, pois aumenta a eficiência do uso de recursos e no desenvolvimento de produtos em larga escala, além de propiciar a países como o Brasil, uma integração em cadeias globais de valor.

O conceito de Indústria 4.0 é representado pela automação industrial e a integração de diferentes tecnologias como inteligência artificial, robótica, internet das coisas e computação em nuvem com o objetivo de promover a digitalização das atividades industriais melhorando os processos e aumentando a produtividade. Os benefícios alcançados com a implantação da indústria 4.0 são muitos. O uso das tecnologias digitais na indústria permitiu aumentar em 22%, em média, a capacidade produtiva de micro, pequenas e médias empresas dos segmentos de alimentos e bebidas, metalmecânica, moveleiro, vestuário e calçados (SENAI, 2019).

Indústrias inteligentes não são apenas aquelas que possuem um alto grau de automação, mas sim aquelas onde os equipamentos e produtos precisam comunicar-se entre si e desenvolver atividades de forma autônomas, tomando decisões sem intervenção de operadores. Parece até uma história retirada de filmes e livros de ficção científica, mas em algumas indústrias automobilística isto já está acontecendo. São lugares que todo o processo está sendo executado sem interferência humana, onde o operador apenas supervisiona o processo através da tela de computadores (COSTA; STEFANO, 2014).

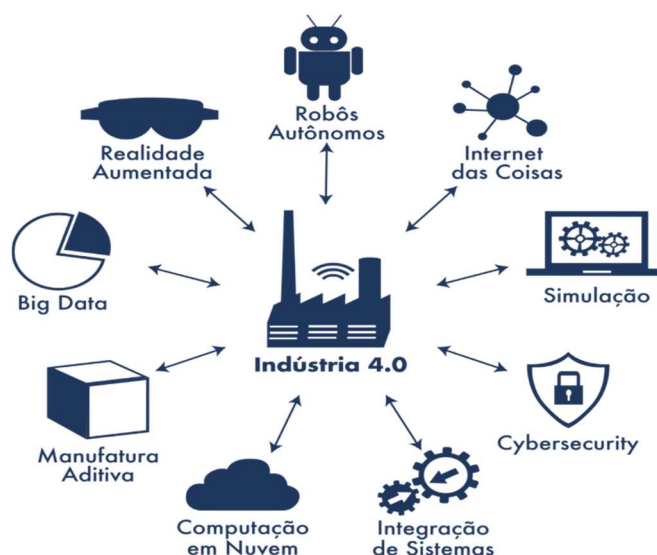
Muitos ainda acreditam que falar de indústria 4.0 é falar de ferramentas complexas, extremamente caras, e que somente grandes empresas com atuação internacional têm acesso ao novo modo de produção. Por meio do programa-piloto Indústria Mais Avançada, do Serviço Nacional de Aprendizagem Industrial (SENAI), executado entre maio de 2018 e outubro de 2019 em 43 empresas de 24 estados, constatou-se o impacto na produção do mercado com o uso de ferramentas de baixo custo, como: sensoriamento, computação em nuvem e Internet das Coisas. A conclusão do programa mostra que o ganho de produtividade está mais relacionado com o quanto se aprende com o processo produtivo, e como esse aprendizado se transforma em ações concretas. As microempresas foram as que mais se beneficiaram do uso inicial de tecnologias digitais. (PORTAL DA INDUSTRIA, 2022).

2.2 Os pilares da indústria 4.0

A indústria 4.0 possui nove pilares que serão o seu apoio, estes são na realidade tecnologias que já estão em uso e a sua introdução no sistema produtivo fará com que se alcance o padrão 4.0 (ESTÉVEZ, 2016). Estes pilares são respectivamente, Robôs Autônomos, Internet da Coisas (IoT), Simulações Computacionais, Cyber segurança, Integração de Sistemas, Computação em Nuvem, Manufatura Aditiva, Big Data e Realidade Aumentada.

Os pilares citados podem ser visualizados na Figura 2 a seguir, sendo que os pilares fundamentais para a realização deste trabalho serão detalhados mais a diante.

Figura 2 – Pilares da Indústria 4.0.



Fonte: Doutor IoT (2022)

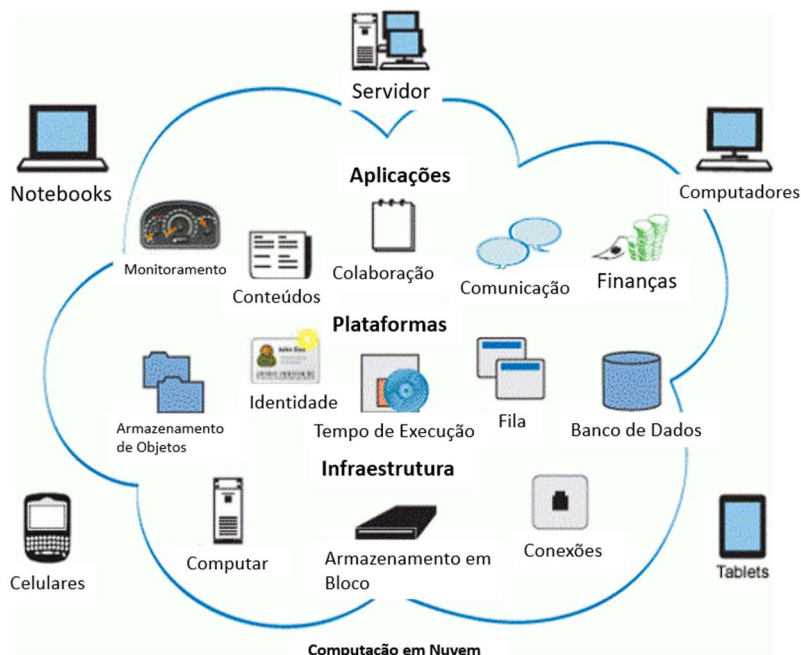
2.2.1 Computação em Nuvem

Computação em Nuvem, ou *Cloud Computing*, é a distribuição de serviços de computação (servidores, armazenamento, bancos de dados, redes, software, análises, inteligência) pela Internet, com utilização de memória, capacidade de armazenamento e cálculo de computadores e servidores hospedados em Datacenter, proporcionando recursos flexíveis e economia na escala.

Com a computação em nuvem precisa-se apenas de um browser no dispositivo (Computador, Tablet ou smartphone) e desta forma pode-se ter acesso a seus arquivos ou aplicativos de qualquer lugar (MICROSOFT, 2016). Desta forma evitam-se investimentos altos em equipamentos e equipe de suporte, permitindo a empresas focarem seus investimentos nas suas atividades principais.

O termo nuvens é utilizado porque os detalhes técnicos e a infraestrutura por trás dos serviços de TI são invisíveis para o usuário final, conforme pode ser visto na Figura 3. Os responsáveis por controlar e manter toda essa infraestrutura são os provedores. (MICROSOFT, 2016).

Figura 3 – Computação em Nuvem.



Fonte: Adaptado de iMasters (2016)

As indústrias do futuro estão aderindo as ferramentas de computação em nuvem para melhorar o gerenciamento dos seus processos através de um ambiente colaborativo, incentivando o compartilhamento de informações e reutilização de código fonte. Nestes ambientes permite-se a criação, simulação e teste de novos produtos (GE, 2016).

Não somente grandes indústrias estão aproveitando os benefícios da computação em nuvens, as pequenas estão descobrindo suas vantagens e com isto estão melhorando seus resultados e se mantendo competitivas no mercado. No que se refere a inovação elas estão conseguindo desenvolver produtos com o uso de plataformas de desenvolvimento antes só acessível a grandes indústrias devido ao seu alto custo de licença (OLIVEIRA, 2015).

2.2.2 Big Data

Para deixar claro, Big Data é nada mais que o nome atribuído a todo grande volume de dados que necessita de tipos específicos de ferramentas e de tecnologias para ser coletado, tratado e analisado de forma adequada para que possa proporcionar a companhia, realizar decisões assertivas.

Uma das principais caracterizações da Indústria 4.0 é permitir a customização em massa e a massificação da personalização na linha de produção. Para que isso seja possível, um novo modelo de tomada de decisões entra em cena, baseado na conexão completa e total do processo produtivo através da distribuição contínua dos dados fornecidos pelos equipamentos via internet, tudo graças a um conceito que ainda será abordado neste trabalho, a internet das coisas (IoT), permitindo a aquisição de dados de uma forma nunca antes visto, com dados em alto volume, alta velocidade e grade variedade.

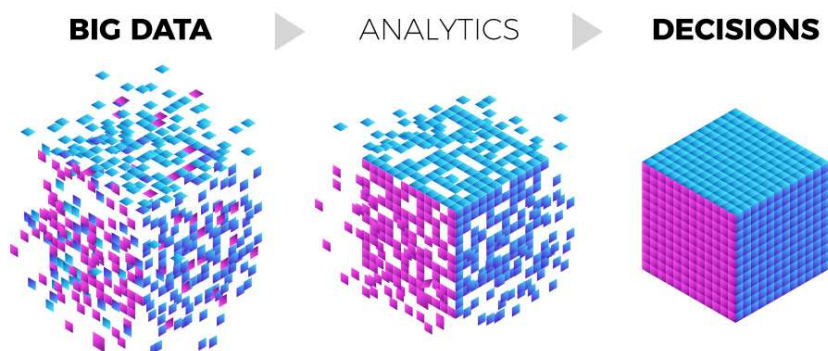
A evolução nas tomadas de decisões na indústria mostra que no início, os primeiros controles apenas apoiavam a tomada de decisões do operador, uma vez que ele é quem tinha o conhecimento do processo e atuava diretamente, logo todo o conhecimento era dele.

Com a evolução das redes, podemos agora gravar estes dados, analisá-los, e tomar decisões baseado em informações e análise local do processo. Ainda é necessário o conhecimento do operador e sua experiência no processo, todavia, as indústrias estão vivendo uma grande questão que é a grande capacidade de aquisição e armazenamento de dados. Isso é algo que já existe, mas não sabem o que fazer com todas estas informações.

Agora, com a Indústria 4.0, o Big Data dá um novo formato a estas informações através das redes convergentes. O sistema aprende conforme o processo ocorre, não há o meio da informação e o resultado é a tomada de decisões em tempo real, com dados relacionados fora no processo local, enxergando toda a cadeia de produção e do negócio.

Com essa tecnologia uma fábrica pode, por exemplo, coletar dados de todas as máquinas, robôs, operadores, transações comerciais com varejo, vendas e transformá-las em informações valiosas, como ilustra a Figura 4.

Figura 4 – Conceito do Big Data.



Fonte: ATRA Informática (2021)

2.2.3 *Internet das coisas (Internet of Things – IoT)*

A ideia por trás da internet das coisas é fazer a conexão de objetos à rede mundial de computadores, estabelecendo a união das redes dos humanos com as diversas redes de objetos (EVANS, 2011). A variedade de objetos conectados é grande, conforme mostra a Figura 3, e as aplicações podem ser as mais diversas possíveis. Por exemplo, é possível conectar um smartphone à geladeira informando o usuário que está faltando algum item (NETSCAN, 2014). Assim como também é possível indicar ao gerente de manutenção que chegou a hora de realizar uma troca de uma correia que alcançou o seu tempo de operação ou que atingiu um desgaste excessivo (NETSERVICE, 2016).

A internet das coisas forçará a uma mudança no consumo e nos meios de produção, criando tendências e estimulando a criação de novos negócios para suprir a demanda por objetos inteligentes concebidos para auxiliar nas tarefas do dia a dia (FIRJAN, 2016).

Segundo Kang (2016), esse sistema de incorporação é conectado por meio de mecanismos de internet sem fio, sendo descrito em quatro camadas de arquitetura, sendo elas a detecção, a rede, o serviço e a interface de conexão.

Para Liu e Xun (2017), a internet das coisas representa mais que um mecanismo de interligação entre o meio natural e o virtual, sendo destacada pelos autores a agilidade por meio de controle remoto em meio aos produtos, o que pode proporcionar maior otimização nos processos produtivos de uma organização

Para Bagheri (2015), internet das coisas é responsável pela coleta de informações no espaço físico, conectando diversos produtos entre si, diferentemente do Cyber Physical Systems que faz uso de nuvens e mecanismos e sensores para ajustar algo físico a um determinado estado projetado.

2.2.4 Digitalização

A digitalização de processos produtivos na indústria é uma realidade que vem ao encontro com algo que está acontecendo em todas as áreas do mercado: a transformação digital. Esta consiste em um procedimento em que as empresas investem em novas tecnologias para melhorar os fluxos de trabalho. E, claro, está ligada à Indústria 4.0, além de impulsionar uma mudança estrutural nas organizações.

A digitalização consiste em realizar as tarefas da sua indústria por meio de ferramentas digitais. Ela é um dos fatores que dão motricidade à 4ª Revolução Industrial, um movimento que tem como característica o uso de tecnologias como Big Data, Inteligência Artificial, Internet das Coisas (IoT) e computação em nuvem. Dessa forma, permite que toda a cadeia industrial seja conectada e se comunique digitalmente, desde a matéria-prima ao cliente final. Assim, a digitalização de processos produtivos possibilita maior agilidade, produtividade e customização de produtos com o menor custo. Também, impulsiona o ganho de eficiência e a continuidade de negócios na manufatura avançada, além de ter operações integradas.

2.3 Inteligência Artificial, machine learning e deep learning

2.3.1 Inteligência Artificial

Inteligência artificial (IA) se refere amplamente a qualquer comportamento semelhante ao do humano apresentado por uma máquina ou sistema. Na forma mais básica da IA, os

computadores são programados para “imitar” o comportamento humano usando dados extensivos de exemplos anteriores de comportamento similar. Eles podem variar desde reconhecer diferenças entre um gato e um pássaro até realizar atividades complexas em uma fábrica.

Embora suas primeiras formas permitissem que computadores jogassem xadrez com humanos, por exemplo, a IA agora faz parte das nossas vidas diárias. Temos soluções de IA para controle de qualidade, análise de vídeo, conversão de fala em texto (processamento de linguagem natural) e condução autônoma, além de soluções na área de saúde, manufatura, serviços financeiros e entretenimento.

É um conceito amplo e que recebe tantas definições quanto significados diferentes à palavra inteligência. É possível considerar algumas características básicas desses sistemas, como a capacidade de raciocínio (aplicar regras lógicas a um conjunto de dados disponíveis para chegar a uma conclusão), aprendizagem (aprender com os erros e acertos para que no futuro possa agir de maneira mais eficaz), reconhecimento de padrões (tanto padrões visuais e sensoriais, como também padrões de comportamento) e inferência (capacidade de conseguir aplicar o raciocínio nas situações do nosso cotidiano).

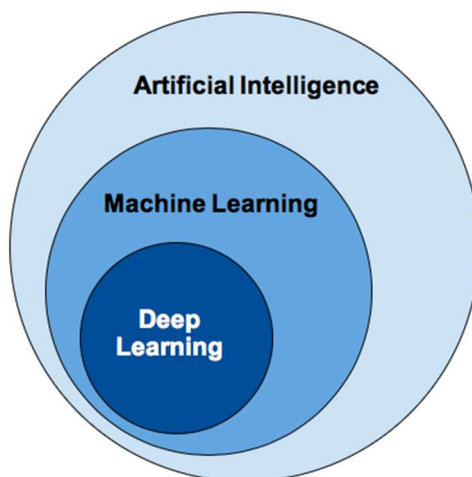
Apenas recentemente, com o surgimento do computador moderno, é que a inteligência artificial ganhou meios e massa crítica para se estabelecer como ciência integral, com problemáticas e metodologias próprias. Desde então, seu desenvolvimento tem extrapolado os clássicos programas de xadrez ou de conversão e envolvido áreas como visão computacional, análise e síntese da voz, lógica difusa, redes neurais artificiais e muitas outras. Inicialmente os modelos de IA visavam reproduzir o pensamento humano. Posteriormente, no entanto, tais modelos abraçaram a ideia de reproduzir faculdades humanas como criatividade, autoaperfeiçoamento e uso da linguagem. Porém, o conceito de inteligência artificial ainda é bastante difícil de se definir. Por essa razão, Inteligência Artificial foi (e continua sendo) uma noção que dispõe de múltiplas interpretações, não raro conflitantes ou circulares.

2.3.2 *Machine Learning*

Algumas tecnologias passam a fazer parte do nosso dia a dia de um jeito tão silencioso que a gente dificilmente percebe. Um exemplo é o machine learning, que pode ser traduzido como aprendizado de máquina ou aprendizagem de máquina, esse é um conceito associado à inteligência artificial, razão pela qual é cada vez mais destacado pela mídia.

Atualmente, muito se discute sobre o aprendizado de máquina, Deep Learning e Inteligência Artificial, acredita-se que esses três termos são sinônimos, mas na verdade não é bem assim. Inteligência Artificial é um conceito amplo que inclui o aprendizado de máquina ou Machine Learning como um dos seus recursos, e o Deep Learning é uma subcategoria do Machine Learning, como mostra a Figura 5.

Figura 5 – Relação entre IA, Machine learning e Deep learning.



Fonte: Venturus (2019)

Machine learning é um ramo da inteligência artificial (IA) e da ciência da computação que se concentra no uso de dados e algoritmos para imitar a maneira como os humanos aprendem, melhorando gradualmente sua precisão. Machine learning é um componente importante do crescente campo da ciência de dados. Por meio do uso de métodos estatísticos, os algoritmos são treinados para fazer classificações ou previsões, revelando os principais insights em projetos de mineração de dados. Esses insights subsequentemente conduzem a tomada de decisões em aplicativos e negócios, impactando de forma ideal as principais métricas de crescimento. Conforme o big data continua a se expandir e crescer, a demanda do mercado por cientistas de dados aumentará, exigindo que eles auxiliem na identificação das questões de negócios mais relevantes e, posteriormente, os dados para respondê-las.

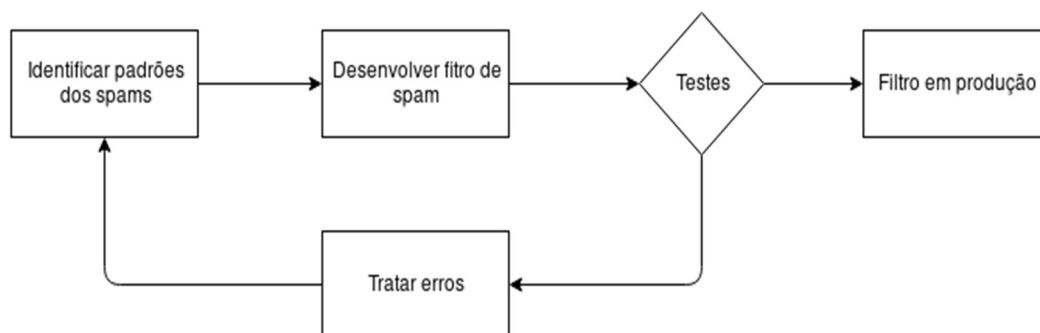
UC Berkeley quebra o sistema de aprendizado de um algoritmo de machine learning em três partes principais:

1. **Um processo de decisão:** Em geral, algoritmos de machine learning são usados para fazer uma predição ou classificação. Com base em alguns dados de entrada, que podem ser rotulados ou não rotulados, seu algoritmo produzirá uma estimativa sobre um padrão nos dados.
2. **Uma função de erro:** Uma função de erro serve para avaliar a predição do modelo. Se houver exemplos conhecidos, uma função de erro poderá fazer uma comparação para avaliar a precisão do modelo.
3. **Um processo de otimização de modelo:** Se o modelo pode se ajustar melhor aos pontos de dados no conjunto de treinamento, então os pesos são ajustados para reduzir a discrepância entre o exemplo conhecido e a estimativa do modelo. O algoritmo repetirá este processo de avaliação e otimização, atualizando os pesos de maneira autônoma até que um limite de precisão seja atingido.

Para melhor compreensão de onde é possível aplicar *Machine Learning*, será utilizado como exemplo um filtro de spam. Tradicionalmente, pode-se escrever um filtro de spam criando regras para identificar padrões que são comuns em e-mails de spam, como palavras específicas presentes no corpo, assunto e até mesmo no campo de remetente. Após a criação do filtro, geralmente realiza-se alguns testes para verificar se o funcionamento está ocorrendo conforme o esperado e caso não esteja, os erros devem ser analisados e as regras reescritas, como mostrado na Figura 6.

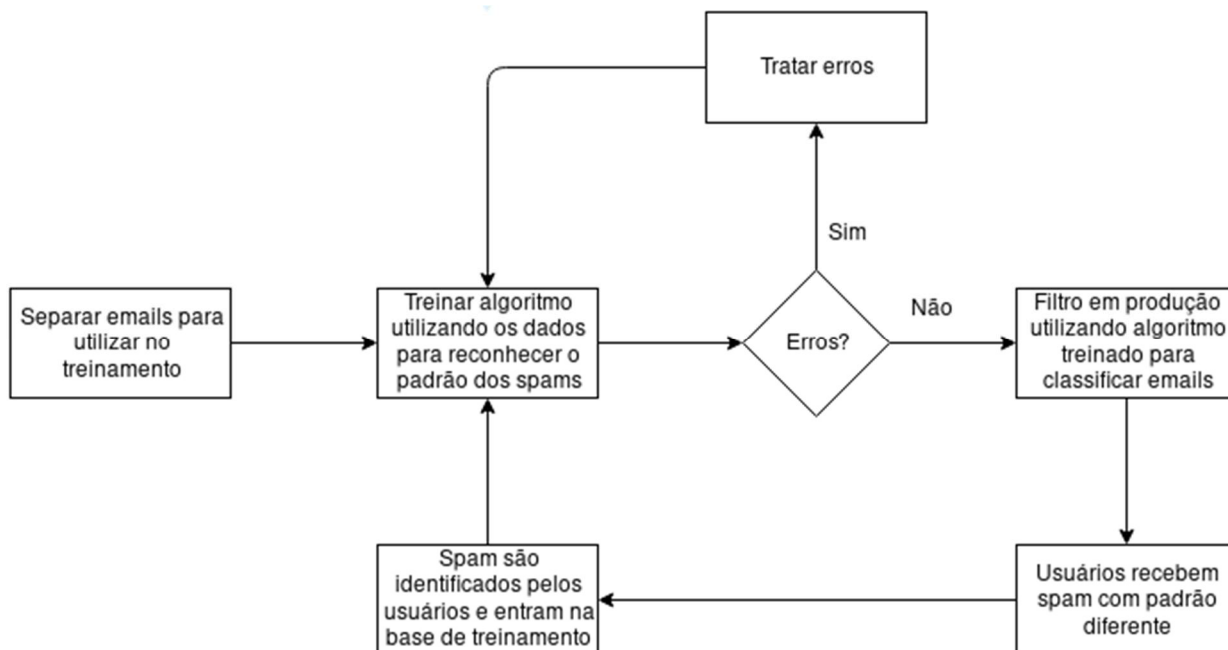
Veja que o parágrafo acima mostra um ciclo de um trabalho manual para que o filtro de spam possa entrar em produção. Porém, mesmo com o filtro funcionando corretamente e em produção, com o passar do tempo, podem surgir novos “padrões” de spam e com isso o ciclo deve se repetir, desde a identificação dos padrões até a execução dos testes.

Figura 6 – Ciclo de implementação manual do filtro de spam.



Com este cenário, temos um trabalho manual que pode ser automatizado e ter até uma eficiência melhor se utilizarmos o Machine Learning, como na Figura 7.

Figura 7 – Utilizando um algoritmo de Machine Learning para classificar os e-mails.

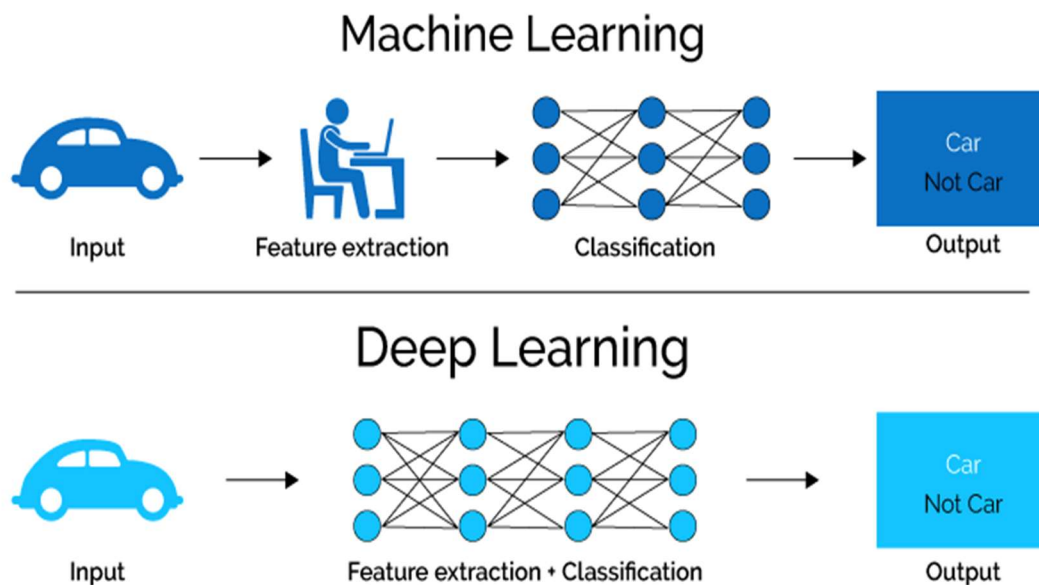


Fonte: 4linux (2019)

2.3.3 *Deep Learning*

Tendo em vista o que é machine learning, temos que Deep Learning é o nome dado a um conjunto de técnicas de redes neurais com múltiplas camadas. Esse tipo de técnica é popular devido a abstrair da necessidade de uma etapa prévia de extração de características. Contrário a outros métodos de Machine Learning, o algoritmo extrai características automaticamente, como mostra a Figura 8.

Figura 8 – Diferença de Machine Learning e Deep Learning.



Fonte: Remiengineering (2019)

Como o conceito de deep learning está diretamente relacionado a redes neurais, o mesmo será melhor descrito nos tópicos subsequentes.

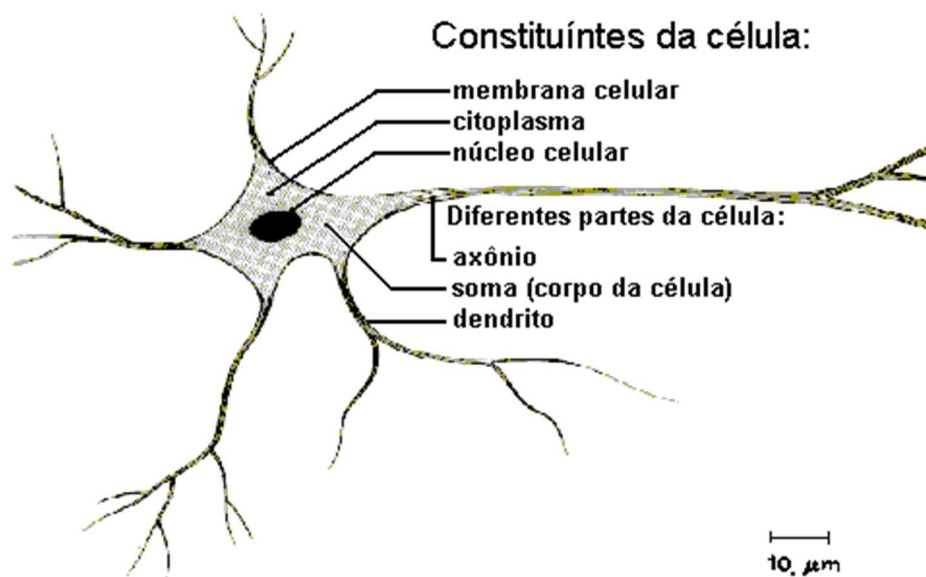
2.4 Redes Neurais

O sistema nervoso é formado por um conjunto extremamente complexo de células, os neurônios. Eles têm um papel essencial na determinação do funcionamento e comportamento do corpo humano e do raciocínio. Os neurônios são formados pelos dendritos, que são um conjunto de terminais de entrada, pelo corpo central, e pelos axônios que são longos terminais de saída.

Os neurônios se comunicam através de sinapses. Sinapse é a região onde dois neurônios entram em contato e através da qual os impulsos nervosos são transmitidos entre eles. Os impulsos recebidos por um neurônio A, em um determinado momento, são processados, e atingindo um dado limiar de ação, o neurônio A dispara, produzindo uma substância neurotransmissora que flui do corpo celular para o axônio, que pode estar conectado a um dendrito de um outro neurônio B. O neurotransmissor pode diminuir ou aumentar a polaridade da membrana pós-sináptica, inibindo ou excitando a geração dos pulsos no neurônio B. Este

processo depende de vários fatores, como a geometria da sinapse e o tipo de neurotransmissor. Como mostra a Figura 9.

Figura 9 – Constituição de uma célula neural.



Fonte: USP (2022)

Uma rede neural artificial (RNA) é composta por várias unidades de processamento (*Nós*), cujo funcionamento é bastante simples. Essas unidades, geralmente são conectadas por canais de comunicação que estão associados a determinado peso. As unidades fazem operações apenas sobre seus dados locais, que são entradas recebidas pelas suas conexões. O comportamento inteligente de uma Rede Neural Artificial vem das interações entre as unidades de processamento da rede.

A maioria dos modelos de redes neurais possui alguma regra de treinamento, onde os pesos de suas conexões são ajustados de acordo com os padrões apresentados. Em outras palavras, elas aprendem através de exemplos.

Uma rede neural é um método de inteligência artificial que ensina computadores a processar dados de uma forma inspirada na estrutura neural de organismos inteligentes e que adquirem conhecimento através da experiência. É um tipo de processo de machine learning, chamado deep learning, que usa nós ou neurônios interconectados em uma estrutura em

camadas, semelhante ao cérebro humano. A rede neural cria um sistema adaptativo que os computadores usam para aprender com os erros e se aprimorar continuamente.

Com a premissa de que se uma máquina tivesse os componentes do cérebro ela funcionaria como tal, o primeiro modelo de neurônio artificial surgiu com McCulloch e Pitts em 1943. O modelo de McCulloch se baseia em um neurônio biológico, cujo funcionamento ocorre por meio da lógica matemática: sinais de entrada multiplicados por pesos sinápticos (representando os dendritos), em que o somatório (corpo celular) desse vetor de entradas produz um valor de saída (axônio), dando funcionamento ao neurônio. Para este modelo “a saída de um neurônio assume o valor 1, se o campo local induzido daquele neurônio é não-negativo, e 0 caso contrário” (HAYKIN, 2001, p.39).

Uma grande rede neural artificial pode ter centenas ou milhares de unidades de processamento, já o cérebro de um mamífero pode ter muitos bilhões de neurônios.

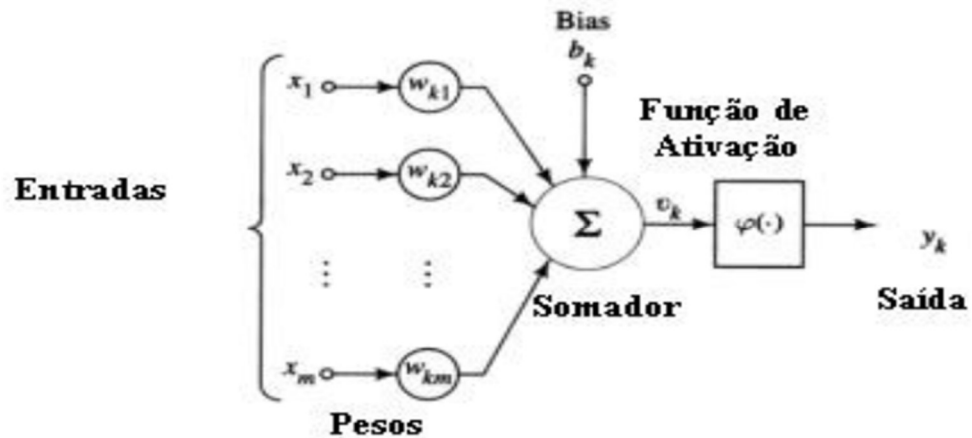
Segundo Wray et. al. (1994), a análise de redes neurais tem três vantagens primárias. A primeira é que o desenvolvimento da rede não necessita do conhecimento das suas relações fundamentais entre as variáveis de entrada e saída. A segunda é que a sua habilidade associativa a torna mais robusta e em condições de trabalhar com situações de perdas ou incertezas dos dados. Adicionalmente, o desempenho das redes neurais não é prejudicado por problemas de multicolinearidade, no qual as variáveis independentes possuem relações lineares exatas ou aproximadamente exatas.

A operação de uma unidade de processamento, proposta por McCulloch e Pitts em 1943, pode ser resumida da seguinte maneira:

- Sinais são apresentados à entrada.
- Cada sinal é multiplicado por um número, ou peso, que indica a sua influência na saída da unidade.
- Um termo de limiar ou Bias, diminui ou aumenta o efeito das entradas.
- É feita a soma ponderada dos sinais que produz um nível de atividade.
- Se este nível de atividade exceder um certo limite (threshold ou iterações) a unidade produz uma determinada resposta de saída.

As descrições anteriores podem ser visualizadas na Figura 10:

Figura 10 – Esquema de unidade McCullock - Pitts.

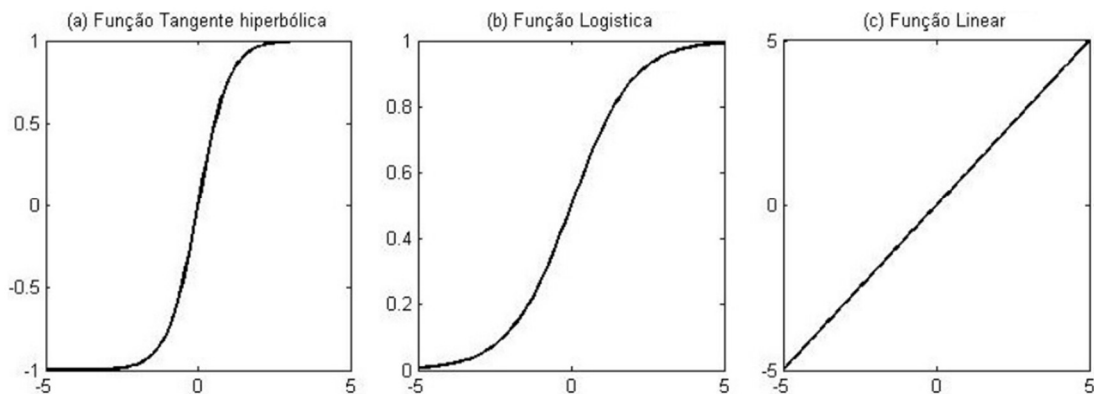


Fonte: Adaptado de Haykin (2001)

Suponha que tenhamos “m” sinais de entrada “ X_1, X_2, \dots, X_m ”, pesos “ W_1, W_2, \dots, W_m ”, “ U_k ” é a saída do combinador linear, “ b_k ” é o polarizador ou bias, “ $\varphi(\cdot)$ ” é a função de ativação e “ Y_k ” é a saída do neurônio.

A Figura 11 apresenta a três funções de ativação comumente utilizadas.

Figura 11 – Funções de Ativação.



Fonte: Adaptado de Haykin (2001)

Estas funções de ativação são escolhidas por sua fácil derivação o que simplifica a aplicação dos algoritmos de aprendizado como será visto a seguir.

Haykin (2001) descreve matematicamente o neurônio artificial da Figura 11 como:

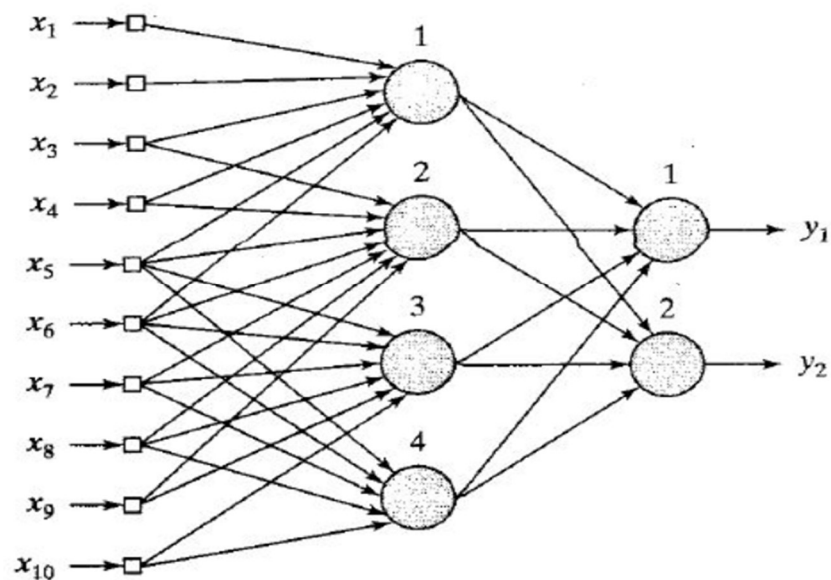
$$U_k = \sum_{j=1}^m W_{kj} X_j$$

$$Y_k = \varphi(v_k)$$

$$v_k = U_k + b_k$$

Segundo Braga (2007) a capacidade de processamento de um neurônio é muito limitada, no entanto quando neurônios são agrupados a capacidade de processamento de dados aumenta. Existem diversos modelos de RNA, que variam de acordo com a forma que os neurônios são organizados e o tipo de algoritmo de aprendizado utilizado. Arquiteturas neurais são tipicamente organizadas em camadas, com unidades que podem estar conectadas às unidades da camada posterior. Na Figura 12, tem-se uma representação deste tipo de arquitetura.

Figura 12 – Arquitetura de uma rede neural.



Fonte: Adaptado de Haykin (2001)

Usualmente as camadas são classificadas em três grupos:

- **Camada de Entrada:** onde os padrões são apresentados à rede.
- **Camadas Intermediárias ou Escondidas ou Ocultas:** onde é feita a maior parte do processamento, através das conexões ponderadas; podem ser consideradas como extratoras de características.

- **Camada de Saída:** onde o resultado final é concluído e apresentado.

A propriedade mais importante das redes neurais é a habilidade de aprender de seu ambiente e com isso melhorar seu desempenho. Isso é feito através de um processo iterativo de ajustes aplicado a seus pesos, o treinamento. O aprendizado ocorre quando a rede neural atinge uma solução generalizada para uma classe de problemas.

Denomina-se algoritmo de aprendizado a um conjunto de regras bem definidas para a solução de um problema de aprendizado. Existem muitos tipos de algoritmos de aprendizado específicos para determinados modelos de redes neurais, estes algoritmos diferem entre si principalmente pelo modo como os pesos são modificados.

Outro fator importante é a maneira pela qual uma rede neural se relaciona com o ambiente. Nesse contexto existem os seguintes paradigmas de aprendizado:

- **Aprendizado Supervisionado**, quando é utilizado um agente externo que indica à rede a resposta desejada para o padrão de entrada;
- **Aprendizado Não Supervisionado** (auto-organização), quando não existe um agente externo indicando a resposta desejada para os padrões de entrada;
- **Reforço**, quando um crítico externo avalia a resposta fornecida pela rede.

Denomina-se ciclo uma apresentação de todos os “N” pares (entrada e saída) do conjunto de treinamento no processo de aprendizado. A correção dos pesos num ciclo pode ser executada de dois modos:

1) Modo Padrão: A correção dos pesos acontece a cada apresentação à rede de um exemplo do conjunto de treinamento. Cada correção de pesos baseia-se somente no erro do exemplo apresentado naquela iteração. Assim, em cada ciclo ocorrem “N” correções.

2) Modo Batch: Apenas uma correção é feita por ciclo. Todos os exemplos do conjunto de treinamento são apresentados à rede, seu erro médio é calculado e a partir deste erro fazem-se as correções dos pesos.

Atualmente, entre as redes neurais mais amplamente utilizadas estão as profundas, as convolucionais, as recorrentes, os autoencoders e as generativas. Mas o universo de

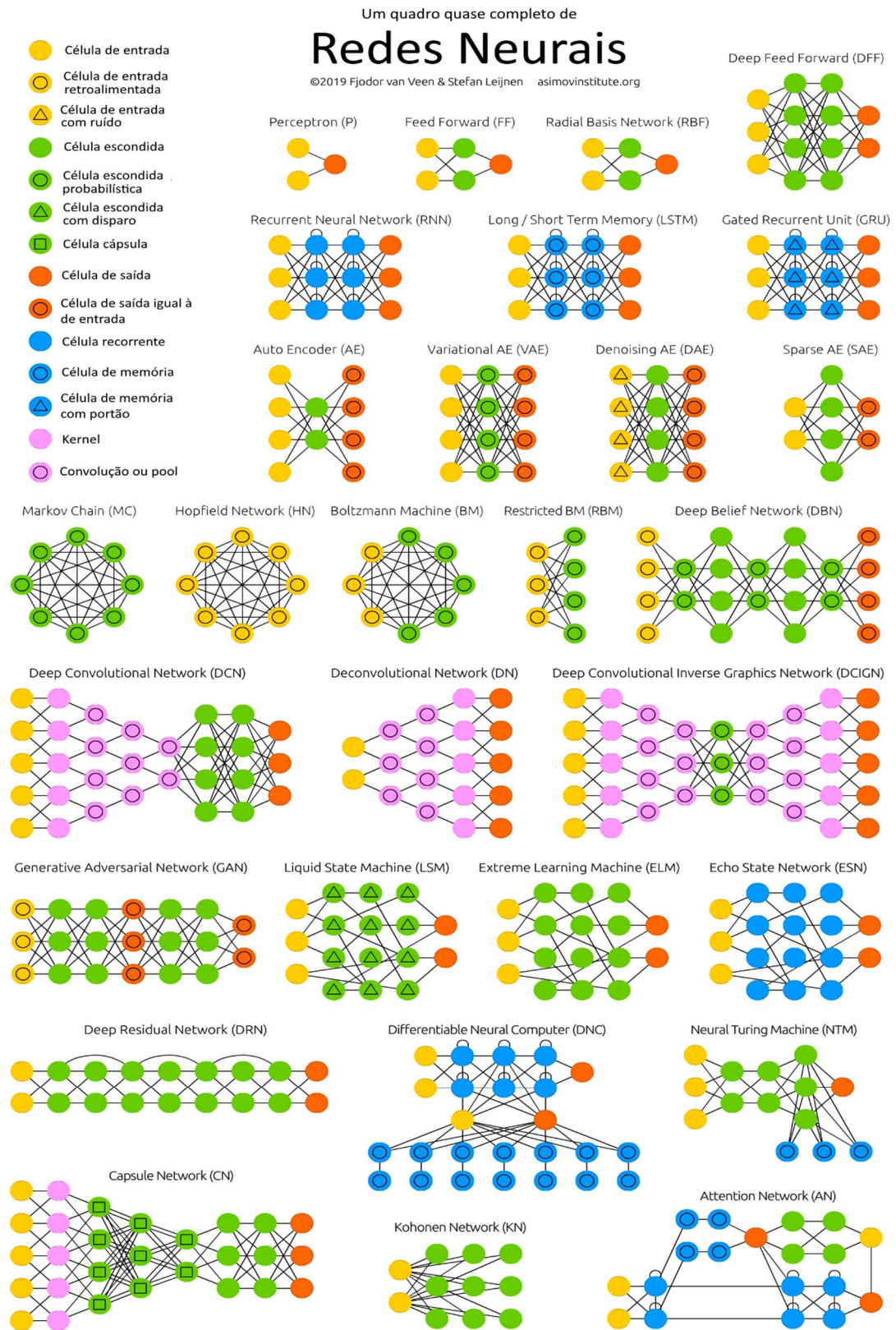
possibilidades é amplo, e várias outras arquiteturas especializadas em tarefas específicas vem sendo bem estabelecidas.

Os principais tipos de redes neurais disponíveis ao desenvolvedor em inteligência artificial possuem características específicas para cada situação, podemos começar com um quadro geral que serve de resumo ao tópico, e depois explicitamos os tipos, suas principais características e aplicações.

2.5 Tipos de Redes Neurais

A seguir, consta um quadro geral, com todos os tipos de redes neurais desenvolvidos até o momento, tendo em vista a grande variedade, serão descritos apenas aqueles cujo suas características são dignas de pauta, pois levantaram pontos importantes para a construção e entendimento deste trabalho.

Figura 13 – Quadro geral dos tipos de redes neurais.



Fonte: Adaptado de Iaexpert (2020)

2.5.1 *Perceptron (P) e Feed Forward (FFN)*

São o tipo mais básico, onde a informação de entrada flui em sequência linear até a saída, como na Figura 14. Em cada neurônio ocorre uma operação matemática linear do tipo $Wx + b$, onde x é o valor do dado de entrada, e W e b são os parâmetros *peso* e *bias* do neurônio.

Figura 14 – Diagrama do Perceptron e Feed Forward.



Fonte: Adaptado de Iaexpert (2020)

FFNs são capazes de modelar vários problemas onde os dados de entrada têm um impacto atemporal nos dados de saída. Um exemplo é usar informações de um exame de sangue para determinar a presença de uma doença.

2.5.2 *Recurrent Neural Network (RNN)*

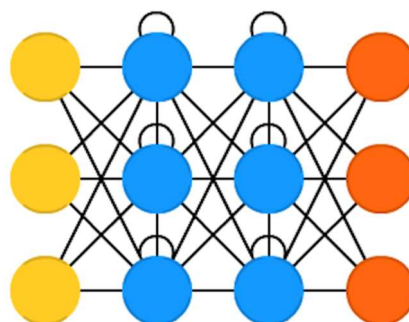
“Os humanos não começam a pensar do zero a cada segundo”, ao ler este texto, entende-se cada palavra com base em sua compreensão das palavras anteriores. Você não joga tudo fora e começa a pensar de novo, pois seus pensamentos têm persistência.

As redes neurais artificiais tradicionais não podem fazer isso, o que traz algumas limitações para esses tipos de modelos. Por exemplo, imagine que você queira classificar o tipo de evento que está acontecendo em todos os pontos de um filme. Não está claro como uma rede neural tradicional poderia usar seu raciocínio sobre eventos anteriores no filme para informar os posteriores. Redes Neurais Recorrentes, ou RNN's, resolvem esse problema, pois são redes com loops, permitindo que as informações persistam.

As redes recorrentes são um tipo de rede neural artificial projetada para reconhecer padrões em sequências de dados, como texto, genomas, caligrafia, palavra falada ou dados de séries numéricas que emanam de sensores, bolsas de valores e agências governamentais. Esses algoritmos consideram tempo e sequência, eles têm uma dimensão temporal, juntamente com outros tipos, como LSTM e GRU.

Como as redes recorrentes possuem um certo tipo de memória, e a memória também faz parte da condição humana, faremos analogias com a memória do cérebro, para uma melhor compreensão. Na Figura 15, é mostrado a arquitetura destes tipos de RN.

Figura 15 – Diagrama da RNN.

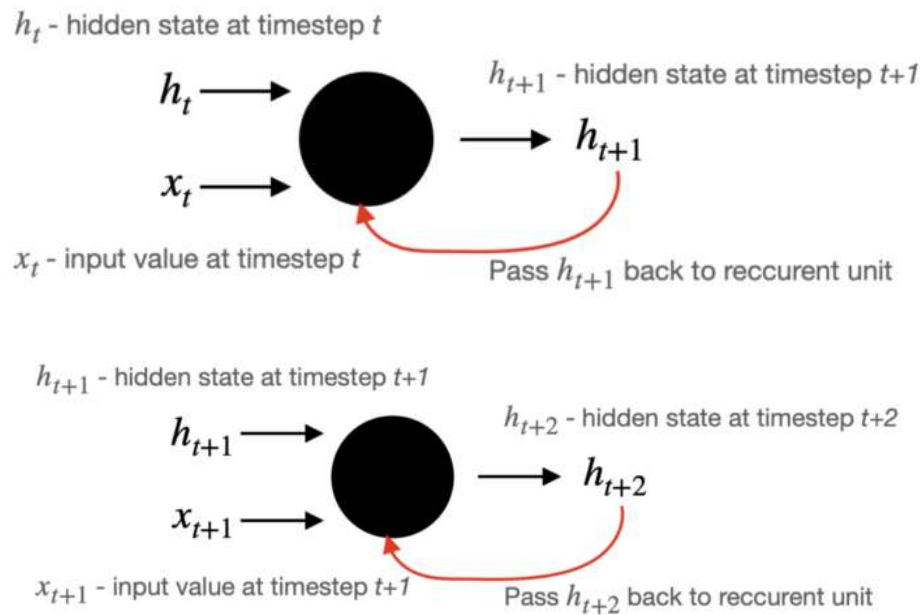


Fonte: Adaptado de Iaexpert (2020)

A decisão de uma rede recorrente alcançada na etapa de tempo $t-1$ afeta a decisão que alcançará um momento mais tarde na etapa de tempo t . Assim, as redes recorrentes têm duas fontes de entrada, o presente e o passado recente, que se combinam para determinar como respondem a novos dados, da mesma forma que fazemos na vida.

As redes recorrentes são diferenciadas das redes feed forward pelo loop de feedback conectado às suas decisões anteriores, ingerindo suas próprias saídas momento após momento como entrada. Costuma-se dizer que as redes recorrentes têm memória. A adição de memória às redes neurais tem uma finalidade: há informações na própria sequência e as redes recorrentes a utilizam para executar tarefas que as redes de feed forward não conseguem. A Figura 16, ilustra este processo de recorrência.

Figura 16 – Representação do processo de recorrência nos nós das camadas ocultas.



Fonte: Adaptado de Iaexpert (2020)

Essa informação sequencial é preservada no estado oculto da rede recorrente, que consegue passar por muitas etapas de tempo à medida que ela avança em cascata para afetar o processamento de cada novo exemplo. Essas correlações entre eventos são separadas por muitos momentos, e essas correlações são chamadas de “**dependências de longo prazo**”, porque um evento no tempo depende e é uma função de um ou mais eventos que vieram antes. Uma maneira objetiva de pensar sobre as RNNs é a seguinte: elas são uma forma de compartilhar pesos ao longo do tempo.

Descreveremos o processo de levar a memória adiante matematicamente da seguinte forma:

Equação 01 – Estado oculto dos nós, no tempo.

$$\mathbf{h}_t = \phi(W\mathbf{x}_t + U\mathbf{h}_{t-1}),$$

O estado oculto na etapa de tempo “ t ” é “ h_t ”. É uma função da entrada na mesma etapa de tempo “ x_t ”, modificada por uma matriz de peso “ W ” (como a que usamos para redes feedforward) adicionada ao estado oculto do passo de tempo anterior “ h_{t-1} ” multiplicado por seu próprio estado

oculto – para a matriz de estado oculto “ U ”, também conhecida como matriz de transição e semelhante a uma cadeia de Markov. As matrizes de peso são filtros que determinam quanta importância deve ser dada tanto à entrada atual quanto ao estado oculto do passado. O erro que eles geram retornará por meio de retropropagação (backpropagation) e será usado para ajustar seus pesos até que o erro não diminua mais.

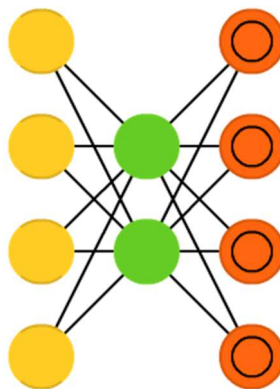
A soma da entrada de peso e do estado oculto é comprimida pela função ϕ – ou uma função sigmoide logística ou tanh, dependendo – que é uma ferramenta padrão para condensar valores muito grandes ou muito pequenos em um espaço logístico, bem como tornar os gradientes viáveis para retropropagação.

Como esse loop de feedback ocorre a cada etapa da série, cada estado oculto contém traços não apenas do estado oculto anterior, mas também de todos aqueles que precederam “ $ht-1$ ” pelo tempo que a memória pode persistir.

2.5.3 *Auto-encoder (AE)*

Os auto-encoders são projetados para representar a informação de entrada em um espaço dimensional menor. Por isso, a camada escondida central dessa rede, que representa esse espaço, deve ter menos neurônios que a camada de entrada. A camada de saída é uma cópia da informação de entrada, de forma que, durante o treinamento, os auto-encoders aprendem a representar a informação original em menos espaço, mas com informação suficiente para reconstruir os dados originais. Como está apresentado na Figura 17.

Figura 17 – Diagrama da AE.



Fonte: Adaptado de Iaexpert (2020)

Os auto-encoders podem ser usados tanto para compactar dados para armazenamento e/ou transmissão, quanto para que representem os dados em forma reduzida para que, por exemplo, outra rede neural especializada em uma tarefa específica possa utilizá-los.

2.6 Séries Temporais

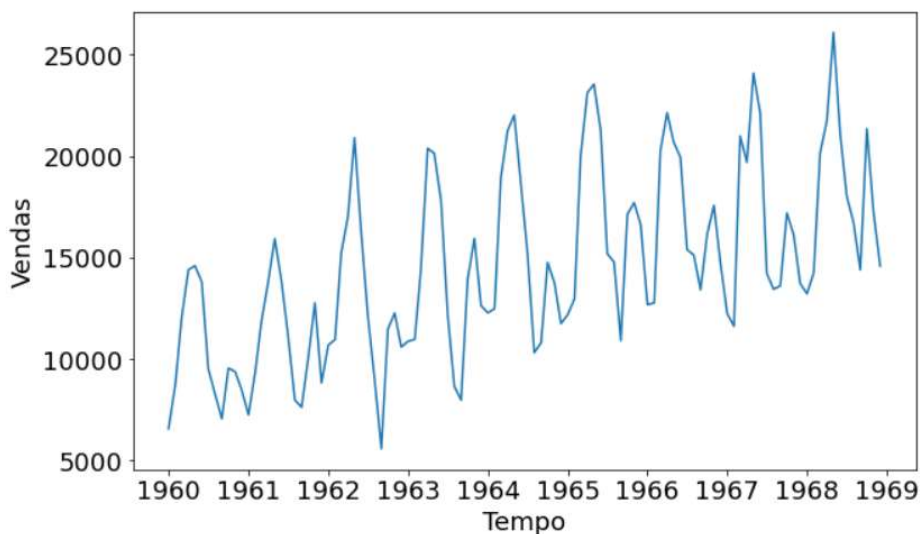
A análise de séries temporais é um importante instrumento no entendimento do mercado e na formulação de planos de ação e estratégias. O histórico de uma variável pode ser utilizado na identificação de períodos de crescimento/decrescimento, sazonalidade e ainda para “prever” observações futuras.

Na verdade, os modelos estatísticos para séries temporais utilizam o passado histórico da variável para projetar observações futuras. Dessa forma, se pode ter uma ideia, em média, de como a variável se comportará nos próximos períodos.

Os modelos estatísticos mais conhecidos e utilizados, como a Regressão Linear e os Modelos Lineares Generalizados (GLM), são adequados na modelagem de variáveis em que as observações são independentes. Em uma série temporal, não há como desconsiderar a estrutura de dependência das observações.

Na Figura 18, pode-se perceber como uma série temporal é descrita.

Figura 18 – Série Temporal.



Fonte: Alura (2021)

Um dos objetivos da série temporal é compreender a estrutura da série, o que é de grande importância pois podemos, por exemplo, tentar entender se existe alguma tendência de crescimento de vendas de um determinado produto, entender como esse produto está comportando e poder tomar decisões a partir de insights gerados com a análise dos dados.

2.6.1 Autocorrelação

A autocorrelação é definida como uma observação num determinado instante está relacionada às observações passadas. As observações podem estar auto correlacionadas em diversas ordens. A autocorrelação de primeira ordem caracteriza séries onde uma observação está correlacionada com a observação imediatamente anterior (fevereiro e janeiro, por exemplo). A autocorrelação de segunda ordem caracteriza séries temporais onde uma observação está correlacionada com as observações a 2 unidades de tempo no passado (fevereiro e dezembro, por exemplo).

2.6.2 Tendência

A tendência de uma série temporal é definida como um padrão de crescimento/decrescimento da variável em um certo período de tempo. Existem testes específicos para a identificação da tendência, como o Teste de Wald e o de Cox-Stuart. Entretanto, uma técnica muito utilizada é o ajuste de uma Regressão Linear Simples para a identificação da inclinação da reta de tendência.

2.6.3 Sazonalidade

A sazonalidade pode ser definida como padrões de comportamento que se repetem em específicas épocas do ano. Por exemplo, o número de passageiros que utilizam o transporte aéreo geralmente é maior em períodos de férias escolares do que nos demais meses do ano, fato que pode ser verificado na Figura 18.

2.6.4 Estacionariedade

A estacionariedade é um importante conceito na modelagem de séries temporais e é caracterizada por uma variável que se comporta de forma aleatória ao longo do tempo ao redor de uma média constante.

2.6.5 Regressão Linear

Os modelos de regressão linear fazem parte de um conjunto de ferramentas comuns entre economistas e estatísticos cujo foco é a realização de inferências, na maior parte das vezes, causais. A inferência consiste em, a partir de evidências encontradas para uma amostra, realizar generalizações de resultados para a população. Ou, de modo mais simples, há um interesse em verificar a correlação entre duas ou mais variáveis e testar o quanto se pode confiar nas estimativas encontradas. Este relacionamento é representado por um modelo matemático, isto é, por uma equação que associa a variável dependente com as variáveis independentes.

A regressão linear pode ser dividida em dois tipos, a regressão linear simples e a múltipla.

Na regressão linear simples, a relação entre duas variáveis pode ser representada por uma linha reta, criando uma relação direta de causa e efeito. Assim, será possível prever os valores de uma variável dependente com base nos resultados da variável independente. É estabelecida a Equação 02:

Equação 02 – Equação da regressão linear simples.

$$Y = \beta_0 + \beta_1 X + E$$

Onde

- Y é a variável dependente (resposta);
- X é a variável independente (explicativa);
- β_0 é o intercepto em Y. É o valor esperado de Y quando $X = 0$;
- β_1 é a inclinação (taxa de mudança). É o aumento esperado em Y quando X aumenta uma unidade;
- E é um erro aleatório, onde se procuram incluir todas as influências no comportamento da variável Y que não podem ser explicadas linearmente pelo comportamento da variável X.
 $E \sim N(0, \sigma_2)$.

Muitas vezes uma única variável explicativa (preditora) não será capaz de explicar tudo a respeito da variável resposta. Se em vez de uma, forem incorporadas várias variáveis independentes, passa-se a ter uma análise de regressão linear múltipla.

Nesse caso, a Equação estabelecida é

Equação 03 – Equação da regressão linear múltipla.

$$Y = \beta_0 + \beta_1 X_1 + \dots + \beta_k X_k + E$$

Onde

- Y é a variável dependente (resposta);
- X_1, \dots, X_k são as variáveis independentes (explicativas);
- β_0, \dots, β_k são os parâmetros desconhecidos do modelo (a estimar). β_0 é o valor esperado de Y quando todas variáveis são iguais a zero. Determinado β_j (j diferente de 0) representa a variação no Y quando X_j aumenta em uma unidade (mantendo constantes todos X_i com exceção do X_j). Se $\beta_j > 0$, Y aumenta e se $\beta_j < 0$, Y diminui;
- E é um erro aleatório, onde se procuram incluir todas as influências no comportamento da variável Y que não podem ser explicadas linearmente pelo comportamento das variáveis X_1, \dots, X_k . $E \sim N(0, \sigma_2)$.

3 METODOLOGIA

Esse trabalho acadêmico se trata de um estudo de caso, onde o principal foco será a realização da previsão das variáveis de processos, na área de refino de gás nitrogênio, em uma empresa que utiliza este material para a produção de gorduras e margarinas, levando-se em consideração as variáveis de processo de que se tem dados coletados.

O estudo foi desenvolvido com base em uma visita técnica em uma empresa do segmento alimentício, localizada na cidade de Fortaleza, Ceará, onde foi analisada toda a sua linha de produção, bem como os pontos de utilização do N₂, sua usina de produção e os equipamentos que realizam a mensuração da quantidade consumida.

Faz-se necessário esclarecer que não serão divulgados dados reais da companhia, pois o sigilo de mercado deve prevalecer. Durante a realização do projeto os dados coletados foram acrescidos de um valor constante, porém essa ação foi feita de tal forma que não afetasse o comportamento das operações matemáticas que serão realizadas.

Os valores de previsão serão desenvolvidos por meio de dados coletados diretamente de sensores instalados nas linhas produtivas da companhia, onde são armazenados em nuvem e somente podem ser acessados através dos servidores da empresa.

Com a coleta dos dados realizada, os mesmos serão devidamente tratados e organizados, por meio de uma interface digital que se utiliza da linguagem SQL para a realização dos procedimentos e provendo os dados organizados em forma de séries temporais, ideais para as demais operações.

Logo após a etapa de tratamento dos dados, estes serão exportados para uma plataforma de programação computacional, onde será utilizada a linguagem Python para que por meio desta sejam realizados os estudos estatísticos necessários, bem como a construção dos modelos de previsão de dados.

Por fim serão averiguados os resultados fornecidos, bem como sua exatidão, em relação aos valores coletados empiricamente.

3.1 Variáveis de processo

Como já foi descrito anteriormente, a empresa em foco atua no setor de margarinas, a mesma possui como sua principal matéria prima, o óleo vegetal que chega na companhia por meio de navios ou caminhões. Este óleo vegetal é coletado e transportado, por meio de tubulações, para a área de *Tancagem*, onde fica armazenado em grandes tanques fechados e aguardará pela demanda da linha produtiva.

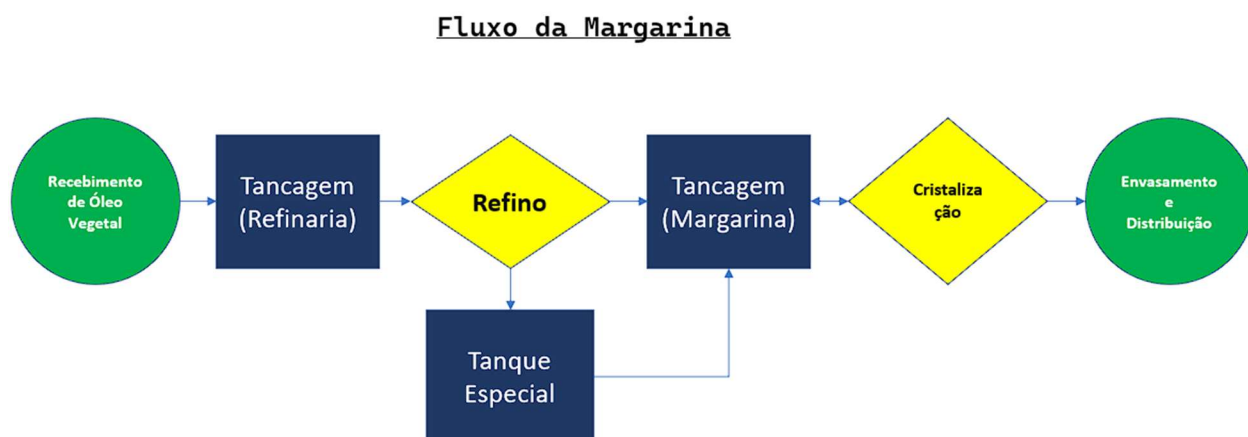
No momento em que for requisitado, a matéria prima é levada da *Tancagem* para a *Refinaria*, onde a mesma passa por processos físico-químicos que permitirão a retirada de impurezas, tendo em vista que os óleos brutos contêm impurezas como ácidos graxos livres, mono e diglicerídios, pigmentos, metais, tocoferóis, esteróis, fosfatídios, ceras, resíduos de farelo, proteínas, açúcares, amido, pesticidas, microtoxinas, umidade, sujeira etc. Portanto, fazem-se necessários processos de desodorização e destilação, que deixarão o produto aceitável para consumo humano.

Após a etapa de refino, o óleo pode ser direcionado para dois locais, dependendo da demanda da linha de produção, estas possibilidades são: I) O óleo é direcionado para um *Tanque Especial* para produto acabado, onde irá aguardar até que o mesmo seja necessário; II) O óleo é diretamente destinado para o setor de *Margarinas*, onde será armazenado em sua área de *Tancagem*, local em que o óleo ficará alternando de lugar, entre os tanques e a área de *Cristalização*.

Na área de *Cristalização*, o material passa por processos de refrigeração que promoverão a formação de cristais na microestrutura do óleo, o que dá a ele a propriedade pastosa da margarina. Após a *Cristalização*, a margarina é devidamente embalada e enviada aos veículos de transporte.

A seguir, na Figura 19, tem-se um fluxograma que busca ilustrar a linha de produção da companhia.

Figura 19 – Linha de Produção da Empresa.



Fonte: Autor (2022)

O gás nitrogênio é utilizado, nesse processo descrito anteriormente, para algumas atividades específicas que garantem o manutenção da continuidade das operações, dentre estas atividades temos: I) Atuar como material inerte que fica em contato com o óleo em todo o processo, substituindo o ar atmosférico que pode conter impurezas que afetem a qualidade do produto, além de impedir a oxigenação do óleo refinado, o que representaria a perda do material e um gasto desnecessário para a empresa; II) Utilizado como fluido de compressão para limpeza interna das tubulações de óleo ou para o acionamento de válvulas e atuadores; III) Por fim, o baixo custo atrelado a sua produção.

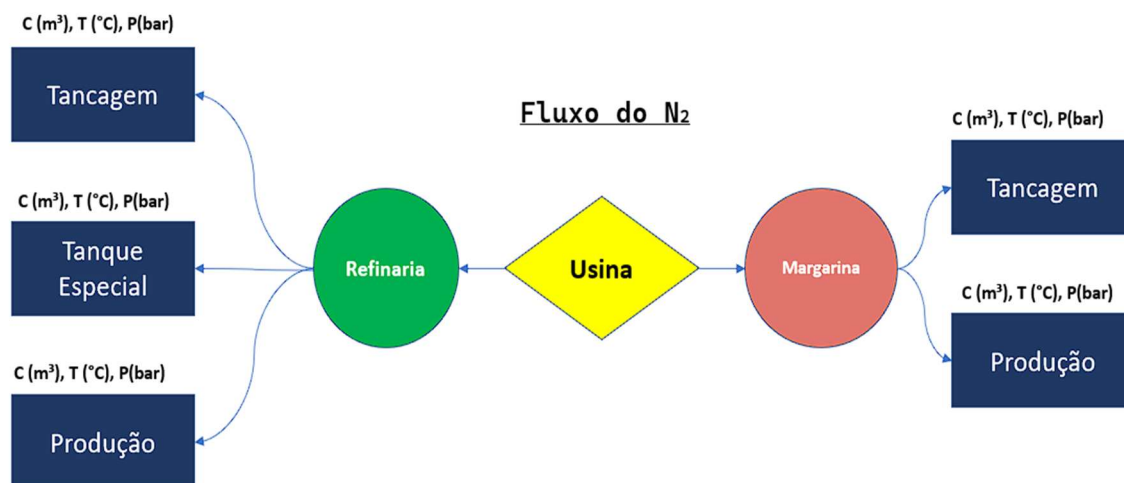
Devido a importância do N₂, a companhia conta com uma usina produtora do mesmo, onde ocorre a separação desse composto, com o ar atmosférico. Esta usina de gás nitrogênio envia-o para todos os pontos de utilização da planta industrial e, à medida que isso ocorre, é feita a coleta dos principais dados referentes as condições desse composto, tudo coletado por meio de sensores e salvo em nuvem para que possa vir a ser monitorado pela fábrica.

Temos entre estas variáveis de condições de utilização:

- O consumo do gás, em metros cúbicos (m³)
- A pressão do gás, em bares (bar)
- A temperatura do gás, em graus celsius (°C)

A fábrica possui valores coletados, para as variáveis descritas, para todas as áreas que possuem requisição do N₂. Na Figura 20 a seguir está ilustrado os principais pontos de utilização do gás nitrogênio.

Figura 20 – Caminho do gás nitrogênio na linha de produção.



Fonte: Autor (2022)

Por estar em contato direto com o óleo a ser refinado, faz-se necessário que o gás nitrogênio esteja sempre com sua temperatura e pressão fixas em uma certa faixa de valores, para que o mesmo não venha a alterar as propriedades físico-químicas do óleo, ocasionando uma perda de qualidade do produto e conseqüentemente uma perda financeira substancial para a empresa.

3.2 Coleta de Dados

Para a criação do modelo de previsão do consumo total de gás nitrogênio, na indústria em estudo, faz-se necessário obter as principais variáveis de processo, descritas anteriormente, de uma forma estruturada e completamente digitalizadas.

Como foi dito, todos os pontos principais de utilização de N₂, possuem um conjunto de sensores, como manômetros e termômetros, que mensuram as condições a cada instante de segundo. Com o intuito de armazenar, digitalmente, estes dados coletados, a companhia utiliza hardwares de uma companhia terceirizada. Estes hardwares, representados na Figura 21, são acoplados diretamente com os sensores da companhia, como ilustra a Figura 22, e possuem a funcionalidade de coletar os valores registrados pelos equipamentos e enviá-los para um banco de dados, que fica localizado na empresa terceirizada, esses dados são disponibilizados a

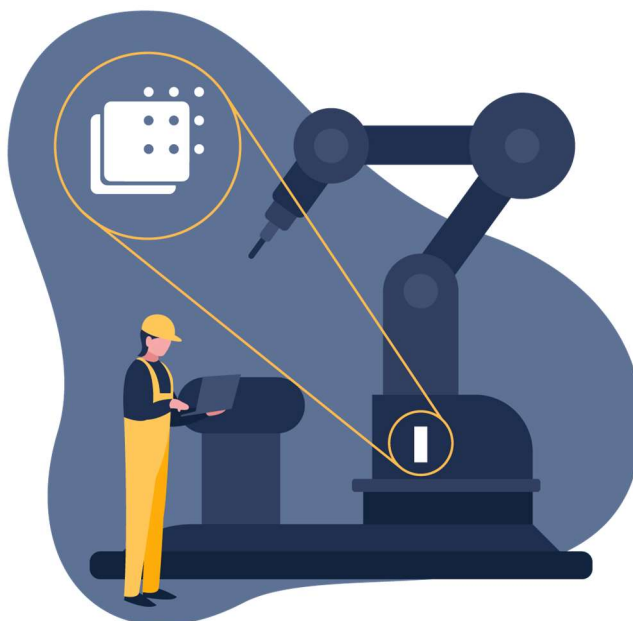
empresa compradora, por meio de uma interface online e de forma completamente organizada, como será melhor descrito a seguir.

Figura 21 – Hardwares utilizados para captação dos dados dos equipamentos.



Fonte: Autor (2022)

Figura 22 – Ilustração da alocação os hardwares.



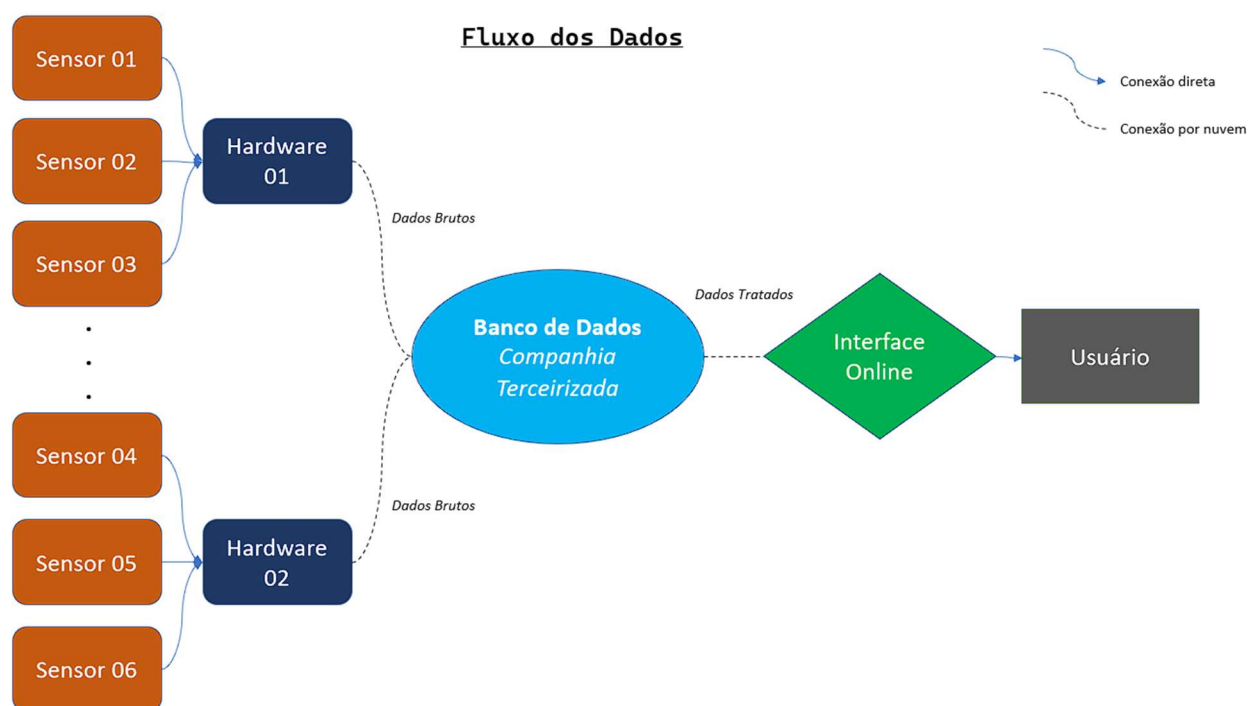
Fonte: St-one (2022)

O fornecimento dos dados via empresa terceirizada, acontece por meio de Tabelas, onde ficam registrados os valores e que se atualizam à medida que novas medições são realizadas.

No entanto, visto que é fácil cometer erros no meio de uma imensidão de dados, faz-se necessário criar um sistema de rótulos para cada dado fornecido por cada sensor.

Para explicar de uma forma melhor, utilizaremos o fluxograma da Figura 23 para descrever todo o caminho percorrido pelos dados fornecidos pela companhia, nele podemos perceber que como já foi falado, os dados são transportados, pelo hardware, dos sensores para a companhia terceirizada em forma de *Dados Brutos*, a estrutura dos *Dados Brutos* será exemplificada pela Tabela 01. Estes dados são tratados pela terceira, e ficam à disposição da companhia de estudo, de forma estruturada, onde é categorizado cada tipo de variável e o seu local de destino. A tabela onde esses *Dados Tratados* podem ser coletados, chama-se *Event* e está exemplificada na Tabela 02.

Figura 23 – Fluxo dos dados.



Fonte: Autor (2022)

Tabela 01 – Exemplo de Tabela de Dados Brutos.

Nº	Sensor	Valor
1	Sensor 01	13
2	Sensor 04	0,5
...
3	Sensor 06	485,33

Fonte: Autor (2022)

Tabela 02 – Exemplo de Tabela de Dados Tratados.

Tempo	Tag	Valor
10/08/2022 22:03:54	tanq_esp.Pressure	13
10/08/2022 22:03:55	prod_ref.Temperature	23,5
...
15/08/2022 09:09:23	prod_ref.Totalizer	485,33

Fonte: Autor (2022)

Por fim, a Tabela 03 demonstra o significado de cada *Tag*.

Tabela 03 – Cadastro de tags.

Tag	Descrição
prod_ref.Totalizer	Consumo de N ₂ de toda a produção da Refinaria
prod_ref.Pressure	Pressão de N ₂ de toda a produção da Refinaria
prod_ref.Temperature	Temperatura do N ₂ de toda a produção da Refinaria
tanc.Totalizer	Consumo de N ₂ de toda a Tancagem
tanc.Pressure	Pressão de N ₂ de toda a Tancagem
tanc.Temperature	Temperatura do N ₂ de toda a Tancagem
tanq_esp.Totalizer	Consumo de N ₂ no Tanque Especial
tanq_esp.Pressure	Pressão de N ₂ no Tanque Especial
tanq_esp.Temperature	Temperatura do N ₂ no Tanque Especial
prod_marg.Totalizer	Consumo de N ₂ de toda a produção da Margarina
prod_marg.Pressure	Pressão de N ₂ de toda a produção da Margarina
prod_marg.Temperature	Temperatura do N ₂ de toda a produção da Margarina
tanc_marg.Totalizer	Consumo de N ₂ de toda a Tancagem da Margarina
tanc_marg.Pressure	Pressão de N ₂ de toda a Tancagem da Margarina
tanc_marg.Temperature	Temperatura de N ₂ de toda a Tancagem da Margarina

Fonte: Autor (2022)

3.3 Tratamento e Extração dos Dados em SQL

Como descrito anteriormente, os dados necessários para a realização do trabalho estão armazenados dentro do banco de dados da companhia terceirizada e necessita de uma interface online adequada, bem como um código de extração na linguagem computacional padrão de banco de dados, a linguagem SQL.

Através de uma interface privada, fornecida pela empresa, os códigos de extração e tratamento de dados foram inseridos, na Figura 24 está uma representação base, da lógica utilizada na escrita SQL.

Figura 24 – Representação Ilustrativa de um código de extração.

```
SELECT $__timeGroup(e.time,'1h',previous) as time, e.tag as "Nome da Coluna Valor", avg(value)
FROM event e
WHERE $__timeFilter(time) AND
      e.tag IN ('Tag da Variavel Coletada')
GROUP BY 1, 2
ORDER BY 1, 2
```

Fonte: Autor (2022)

Onde, temos que os dados a quem forem aplicados este código, fornecerá a média dos valores agrupados a cada 1 hora de extração. Adiante será descrito todo o processo de extração e tratamento dos dados de condição de gás nitrogênio para a área de produção do setor de Refinaria, para um período de 25 de Agosto de 2022, até 24 de Outubro de 2022.

Primeiramente, serão extraídos todos os dados brutos para a tag “prod_ref.Totalizer”, através deste código, da Figura 25:

Figura 25 – Linhas de código do consumo de gás na refinaria.

```
SELECT *
FROM event e
WHERE $__timeFilter(time) AND
      e.tag IN ('prod_ref.Totalizer')
ORDER BY 1, 2
```

Fonte: Autor (2022)

Onde as funções utilizadas são:

- “**SELECT ***”: Seleção de todas as colunas fornecidas pela Tabela selecionada.
- “**FROM event e**”: Seleção da Tabela Evento, descrita anteriormente.
- “**WHERE**”: Função de Aplicação de filtros a Tabela.
- “**\$_timeFilter(time)**”: Filtro aplicado que fornecerá dados apenas no período de interesse.
- “**e.tag IN ('prod_ref.Totalizer')**”: Buscará apenas os valores da tag descrita.
- “**ORDER BY 1, 2**”: Classifica os dados em forma decrescente de tempo.

A Tabela filtrada do banco de dados, nos será apresentada desta forma:

Tabela 04 – Tabela Pré-extraída do banco de dados.

Time	Tag	Value
2022-08-25 08:00:01	prod_ref.Totalizer	17312
2022-08-25 08:00:05	prod_ref.Totalizer	17312
...
2022-10-24 23:30:26	prod_ref.Totalizer	31855

Fonte: Autor (2022)

Pode-se perceber que os dados são adicionados a tabela em períodos muito curtos de tempo, portanto para terminar o processo de tratamento, estes valores serão agrupados nos períodos de 1 hora, como apresentado na Figura 26.

Figura 26 – Linhas de código do consumo de gás na refinaria, agrupado.

```
--CONSUMO PRODUÇÃO REFINARIA--

SELECT $_timeGroup(e.time,'1h',previous) as time, e.tag as Totalizado_N2, avg(value)
FROM event e
WHERE $_timeFilter(time) AND
      e.tag IN ('prod_ref.Totalizer')
GROUP BY 1, 2
ORDER BY 1, 2
```

Fonte: Autor (2022)

Onde:

- “**\$_timeGroup(e.time,'1h',previous) as time**”: Função de agrupamento do tempo.
- “**avg(value)**”: O valor agrupado será a média do consumo.
- “**GROUP BY 1, 2**” Especifica as colunas agrupadas.

Logo, obtemos a Tabela tratada para o consumo de gás nitrogênio:

Tabela 05 – Tabela tratada e extraída do banco de dados.

Time	Totalizado_N2	Value
2022-08-25 08:00:00	prod_ref.Totalizer	17312
2022-08-25 09:00:00	prod_ref.Totalizer	17316
...
2022-10-24 22:00:00	prod_ref.Totalizer	31856
2022-10-24 23:00:00	prod_ref.Totalizer	31858

Fonte: Autor (2022)

Este processo é repetido para todas as outras variáveis da companhia. Por fim, faz-se necessário agregar a esta extração, todas as outras variáveis condicionais de gás nitrogênio, das outras áreas da empresa, esta ação será possível com uma ampliação do código fonte, o que lhe fornecerá maior complexidade. o mesmo está escrito na Figura 27:

Figura 27 – Código de extração completo.

```

1  WITH producao_refinaria_consumo as (
2  SELECT $__timeGroup(e.time,'1h',previous) as time, e.tag as Totalizado_N2,
3  CASE
4  WHEN avg(value) < 0 THEN 0
5  WHEN avg(value) isnull THEN 0
6  ELSE avg(value)
7  END AS "C1"
8  FROM event e
9  WHERE $__timeFilter(time) AND e.tag IN ('prod_ref.Totalizer')
10 GROUP BY 1, 2
11 ORDER BY 1, 2),
12 producao_refinaria_precao as (
13 SELECT $__timeGroup(e.time,'1h',previous) as time, e.tag as Pressao_N2,
14 CASE
15 WHEN avg(value) < 0 THEN 0
16 WHEN avg(value) isnull THEN 0
17 ELSE avg(value)
18 END AS "P1"
19 FROM event e
20 WHERE $__timeFilter(time) AND e.tag IN ('prod_ref.Pressure')
21 GROUP BY 1, 2
22 ORDER BY 1, 2
23 ),
24 producao_refinaria_temperatura as (
25 SELECT $__timeGroup(e.time,'1h',previous) as time, e.tag as Temperatura_N2,
26 CASE
27 WHEN avg(value) < 0 THEN 0
28 WHEN avg(value) isnull THEN 0
29 ELSE avg(value)
30 END AS "T1"
31 FROM event e
32 WHERE $__timeFilter(time) AND e.tag IN ('prod_ref.Temperature')
33 GROUP BY 1, 2
34 ORDER BY 1, 2
35 ),
36
37 juntar01 as (
38 SELECT prc.time as time, prc."C1" as "c1", prp."P1" as "p1"
39 FROM producao_refinaria_consumo prc
40 JOIN producao_refinaria_precao prp ON prc.time = prp.time
41 order by prc.time DESC
42 ),
43 juntar02 as (
44 SELECT j.time as time, prt."T1" as "t1", j."c1", j."p1"
45 FROM juntar01 j
46 JOIN producao_refinaria_temperatura prt ON prt.time = j.time
47 order by j.time DESC
48 )
49
50 SELECT j.time as time,j."c1", j."p1", j."t1"
51 FROM juntar02 j
52 order by j.time DESC

```

Fonte: Autor (2022)

A Tabela de dados, por sua vez, ficará desta forma:

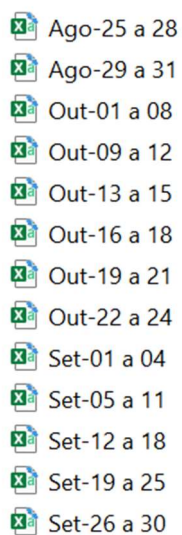
Tabela 06 – Tabela de Dados Completa.

Time	c1	p1	t1
2022-08-25 00:00:00	17333	4,74	27,7
2022-08-25 01:00:00	17335	4,69	27,5
...
2022-10-24 22:00:00	31856	4,66	28,1
2022-10-24 23:00:00	31858	4,72	28,2

Fonte: Autor (2022)

Com este processo concluído, a interface online, da empresa em estudo, nos fornece a opção de exportarmos as tabelas filtradas no formato de planilhas .CSV, do Microsoft Excel, de acordo com a Figura 28.

Figura 28 – Arquivos Extraídos.



Fonte: Autor (2022)

3.4 Análise dos dados via Python

Com a exportação dos dados dos sensores, por meio da interface online da companhia, todas as etapas seguintes serão realizadas por meio da linguagem Python. Vale ressaltar que, será por meio desta linguagem computacional, que será criado o modelo base de nossa rede neural.

3.4.1 *Bibliotecas Utilizadas*

Como etapa inicial do processo de criação de qualquer código computacional, temos que a importação das bibliotecas adequadas para a realização de atividades complexas representa um ponto crucial para o projeto em questão.

Estas bibliotecas, são pacotes de códigos pré-produzidos, com o intuito de agilizar determinados processos, por meio da importação de funções computacionais, que por sua vez, realizaram uma determinada ação. As principais utilizadas neste projeto foram:

- **Tensorflow/Keras:** Biblioteca de deep learning, criada com o intuito de facilitar o desenvolvimento do processo de criação de redes neurais, pois este já possui imbutido em suas funções, os cálculos e equações necessários para a criação das mesmas. Esta é a ferramenta principal do mundo na criação de inteligências artificiais, sendo utilizada em grandes empresas como YouTube, Netflix e NASA.
- **Pandas:** Biblioteca utilizada para operações envolvendo Tabelas e vetores.
- **Numpy e Math:** Bibliotecas para realização de operações matemáticas.
- **Glob:** Biblioteca para operações envolvendo planilhas “.csv”.
- **Sklearn:** Biblioteca criada para análises estatísticas.
- **Plotly, Matplotlib e Seaborn:** Bibliotecas para criação de gráficos.

Na Figura 29, temos o trecho do código referente a importação das bibliotecas citadas:

Figura 29 – Bibliotecas Importadas.

```
# Tensorflow / Keras
from tensorflow import keras # Para construir a rede neural
from keras.models import Sequential # para criar uma pilha linear de camadas para nossa rede neural
from keras import Input # para instanciar um tensor keras
from keras.layers import Dense, SimpleRNN, LSTM, Dropout # para criar camadas NN e camadas RNN regulares densamente conectadas
from keras.optimizers import Adam
# Para Manipulação dos dados
import pandas as pd
import numpy as np
import math # para ajudar na reformulação dos dados
import glob # para unir as planilhas
# Sklearn
import sklearn # para avaliação do modelo
from sklearn.model_selection import train_test_split # para dividir os dados em amostras de treinamento e teste # para métricas de avaliação de modelo
from sklearn.preprocessing import MinMaxScaler, StandardScaler, MaxAbsScaler, RobustScaler # para dimensionamento de recursos
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score # para métricas de avaliação de modelo
# Visualização
import plotly
import plotly.express as px
import plotly.graph_objects as go
import matplotlib.pyplot as plt
import seaborn as sns
```

Fonte: Autor (2022)

3.4.2 Criação de Data Frames

Próximo passo será a importação dos dados coletados pela companhia, para que estes possam ser armazenados na forma de *Data Frames*, ou seja, para que as tabelas presentes nas planilhas sejam incorporadas no formato de tabelas no Python. O código de criação está descrito na Figura 30.

Para isso, será criada uma Tabela chamada “df”, onde serão recebidos todos os valores.

Figura 30 – Criação do Data Frame.

```

# Defina as opções do Pandas para exibir mais colunas
pd.options.display.max_columns=50

path = r'Caminho da Pasta dos arquivos'
filenames = glob.glob(path + "/*.csv")
dfs = []
for filename in filenames:
    dfs.append(pd.read_csv(filename))

# Concatenar todos os arquivos em um dataframe
df = pd.concat(dfs, ignore_index=True)

# Mostrar um instantâneo de dados
df = df[["time", "c1", "p1", "t1"]]
df['time'] = pd.to_datetime(df['time'])
df = df.set_index('time')

```

Fonte: Autor (2022)

A escrita acima nos fornecerá o seguinte *Data Frame* da Figura 31:

Figura 31 – Data Frame Geral.

	c1	p1	t1
time			
2022-08-28 23:00:00	17819	4.74	27.8
2022-08-28 22:00:00	17816	4.64	27.8
2022-08-28 21:00:00	17774	4.02	28.0
2022-08-28 20:00:00	17753	4.74	28.1
2022-08-28 19:00:00	17751	4.62	28.1
...
2022-09-26 04:00:00	24874	4.64	27.5
2022-09-26 03:00:00	24863	4.74	27.5
2022-09-26 02:00:00	24861	4.74	27.5
2022-09-26 01:00:00	24858	4.74	27.6
2022-09-26 00:00:00	24854	4.64	27.7

1456 rows × 3 columns

Fontes: Autor (2022)

Na Figura 32, bem como na Figura 33, podemos ver que, por uma questão de organização e estética, os títulos das colunas foram renomeados de tal forma que fique explícito o significado de cada cadeia de dados.

Figura 32 – Renomeando os Nomes das Variáveis.

```
df = df.rename(columns={'p1': 'Pressão (bar) - Produção Refinaria',
                        't1': 'Temperatura (°C) - Produção Refinaria',
                        'c1': 'Consumo de N2 (m³) - Produção Refinaria'})
```

Fontes: Autor (2022)

Figura 33 – Data Frame Renomeado.

time	Consumo de N2 (m³) - Produção Refinaria	Pressão (bar) - Produção Refinaria	Temperatura (°C) - Produção Refinaria
2022-08-28 23:00:00	17819.0	4.74	27.8
2022-08-28 22:00:00	17816.0	4.64	27.8
2022-08-28 21:00:00	17774.0	4.02	28.0
2022-08-28 20:00:00	17753.0	4.74	28.1
2022-08-28 19:00:00	17751.0	4.62	28.1
...
2022-09-26 04:00:00	24874.0	4.64	27.5
2022-09-26 03:00:00	24863.0	4.74	27.5
2022-09-26 02:00:00	24861.0	4.74	27.5
2022-09-26 01:00:00	24858.0	4.74	27.6
2022-09-26 00:00:00	24854.0	4.64	27.7

1456 rows x 3 columns

Fontes: Autor (2022)

3.4.3 Análise Gráfica

Com os dados, das condições de operação da Refinaria, devidamente coletados e organizados para a plataforma Python, necessita-se realizar uma análise do comportamento das mesmas, para isso, foram plotados os gráficos referentes a todo o período amostral do projeto em questão. Na Figura 34, temos o código de plotagem dos gráficos.

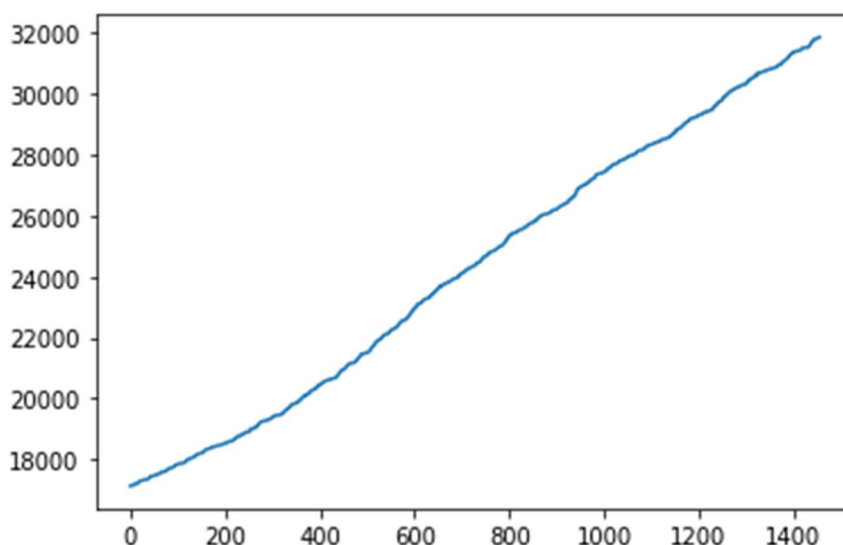
Figura 34 – Plotagem dos dados.

```
df['Consumo de N2 (m³) - Produção Refinaria'].plot()  
df['Pressão (bar) - Produção Refinaria'].plot()  
df['Temperatura (°C) - Produção Refinaria'].plot()
```

Fontes: Autor (2022)

Podemos perceber na Figura 35 que o consumo de gás nitrogênio no setor de destaque, como já esperado, é representado, aproximadamente, por uma função de primeiro grau, tendo em vista que o consumo de gás sempre será maior à medida que for utilizado para os processos e operações da área.

Devido ao seu caráter acumulativo, o valor coletado deverá ser sempre maior ou igual ao seu valor anterior, qualquer valor que esteja fora desta lógica, deverá ser considerado como um *Outlier* e, por sua vez, deverá ser retirado do *Data Frame*.

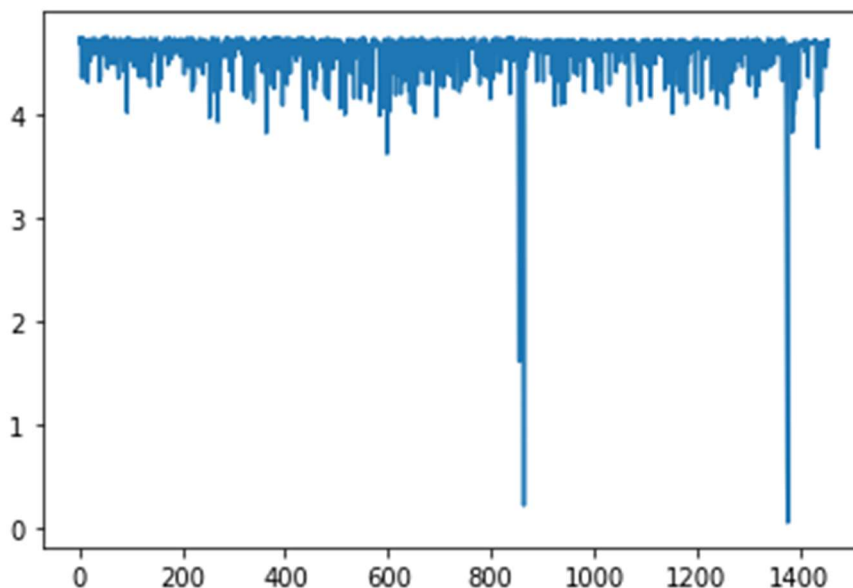
Figura 35 – Consumo de Gás, em m³.

Fontes: Autor (2022)

Juntamente com os dados de consumo, temos que os valores de pressão do gás, na Figura 36, apresentam um comportamento mais imprevisível e com uma maior quantidade de *Outliers*, o que deverá ser devidamente tratado, caso contrário poderá resultar em um empecilho para a criação do modelo de rede neural. Os *Outliers* podem ser causados por

diversos fatores como erros nos equipamentos de medição, ou problemas nas condições de operação.

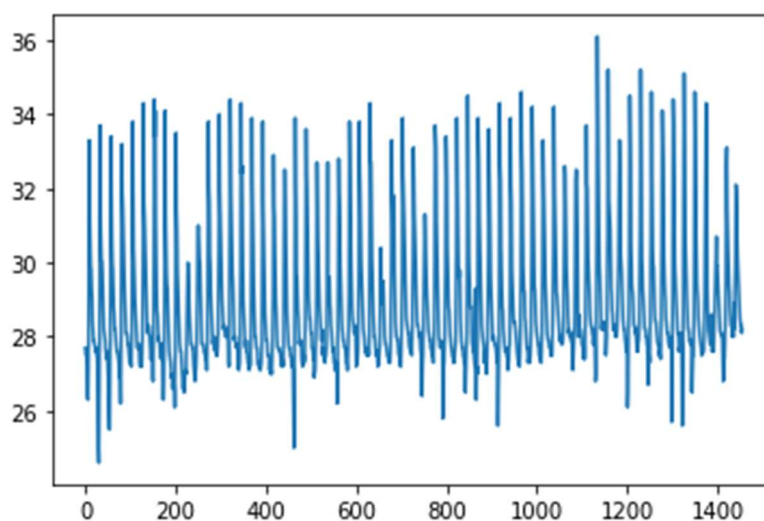
Figura 36 – Pressão de Gás, em bar.



Fontes: Autor (2022)

Por fim, temos que assim como os valores de pressão, os dados de temperatura, da Figura 37, também apresentam uma elevada variação, no entanto percebe-se que esta variável possui um grau de previsibilidade maior do que a pressão, por conta da quantidade de pontos visualmente semelhantes.

Figura 37 – Temperatura de Gás, em °C.



Fontes: Autor (2022)

3.4.4 Análises Estatísticas

Para que o modelo de rede neural possa ser construído de tal forma que apresente o menor erro possível, será necessária uma eliminação de um conjunto de dados, se existirem, que estão fora do comportamento padrão das variáveis. Como foi falado anteriormente, esses dados que destoam dos demais recebem o nome de *Outliers*, e podem ser excluídos das amostras, através da utilização de métodos estatísticos adequados.

Na Figura 38 e 39, consta uma análise do coeficiente de correlação de Spearman entre as 3 amostras do trabalho em questão, esse coeficiente descreve a relação entre as variáveis através de uma função monotética, ou seja, ele está analisando se quando o valor de uma amostra aumenta ou diminui, o valor da outra também se comporta de maneira semelhante. Esse coeficiente de correlação, possui valores que variam de -1 a 1, onde valores próximos de 1 são diretamente proporcionais, além de valores próximos de -1 serem inversamente proporcionais, e por fim, valores próximos de 0 possuem pouca correlação.

Este coeficiente nos indica, através da matriz de correlação da Figura 39, que todas as variáveis possuem uma baixa correlação entre si, o que nos informa como deveremos trabalhar com elas daqui para frente.

Figura 38 – Função do Coeficiente de Correlação de Spearman.

```
correlacoes = df.corr(method='spearman')
```

Fontes: Autor (2022)

Figura 39 – Matriz de correlação das variáveis.

	Consumo de N2 (m ³) - Produção Refinaria	Pressão (bar) - Produção Refinaria	Temperatura (°C) - Produção Refinaria
Consumo de N2 (m ³) - Produção Refinaria	1.000000	-0.036670	0.166336
Pressão (bar) - Produção Refinaria	-0.036670	1.000000	-0.171736
Temperatura (°C) - Produção Refinaria	0.166336	-0.171736	1.000000

Fontes: Autor (2022)

Levando-se em consideração que os dados coletados possuem uma baixa correlação, isso implica que o estudo e desenvolvimento dos modelos de predição e previsão das amostras, deverão ser feitos de forma separadas, ou seja, como uma variável não altera significativamente a outra, não será possível utilizar a mesma variável como auxílio para entender o funcionamento da outra.

Na Figura 40, começamos o processo de eliminação dos *Outliers*. Primeiramente, separamos o nosso *Data Frame* em 3, um para cada amostra. Logo após, plotamos as distribuições das frequências de valores, para cada amostra, e em dois formatos distintos (Histograma e Box Plot).

Os histogramas criados, referentes as Figuras 41, 43 e 45, nos fornecem a percepção de que os estudo em questão está sendo realizado em cima de um conjunto de amostras que não apresentam normalidade. Além dos histogramas, temos nas Figuras 42, 44 e 46, a representação da distribuição em formato box plot (ou diagrama de caixa), onde podemos perceber, nas distribuições de Pressão e Temperatura, os pontos que se encontram fora do comportamento padrão das séries temporais.

Figura 40 – Códigos de histogramas e box plots.

```
# Separação do DF
df_cons = df[["time", "Consumo de N2 (m³) - Produção Refinaria"]]
df_pre = df[["time", "Pressão (bar) - Produção Refinaria"]]
df_temp = df[["time", "Temperatura (°C) - Produção Refinaria"]]

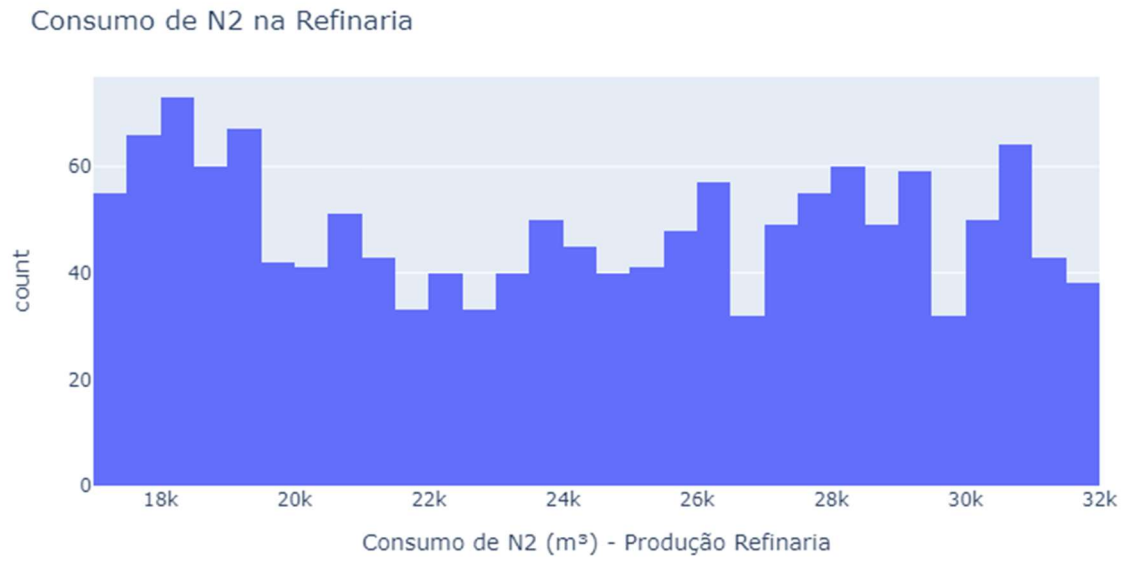
# Histograma e box Plot do Consumo
hist_n2 = px.histogram(df_cons, x = "Consumo de N2 (m³) - Produção Refinaria", nbins=40)
hist_n2.update_layout(width=800, height = 400, title_text="Consumo de N2 na Refinaria")
hist_n2.show()
box_n2 = px.box(df_cons, y="Consumo de N2 (m³) - Produção Refinaria")
box_n2.show()

# Histograma e box Plot da Pressão
hist_n2 = px.histogram(df_pre, x = "Pressão (bar) - Produção Refinaria", nbins=40)
hist_n2.update_layout(width=800, height = 400, title_text="Pressão de N2 na Refinaria")
hist_n2.show()
box_n2 = px.box(df_pre, y="Pressão (bar) - Produção Refinaria")
box_n2.show()

# Histograma e box Plot da Temperatura
hist_n2 = px.histogram(df_temp, x = "Temperatura (°C) - Produção Refinaria", nbins=40)
hist_n2.update_layout(width=800, height = 400, title_text="Temperatura de N2 na Refinaria")
hist_n2.show()
box_n2 = px.box(df_temp, y="Temperatura (°C) - Produção Refinaria")
box_n2.show()
```

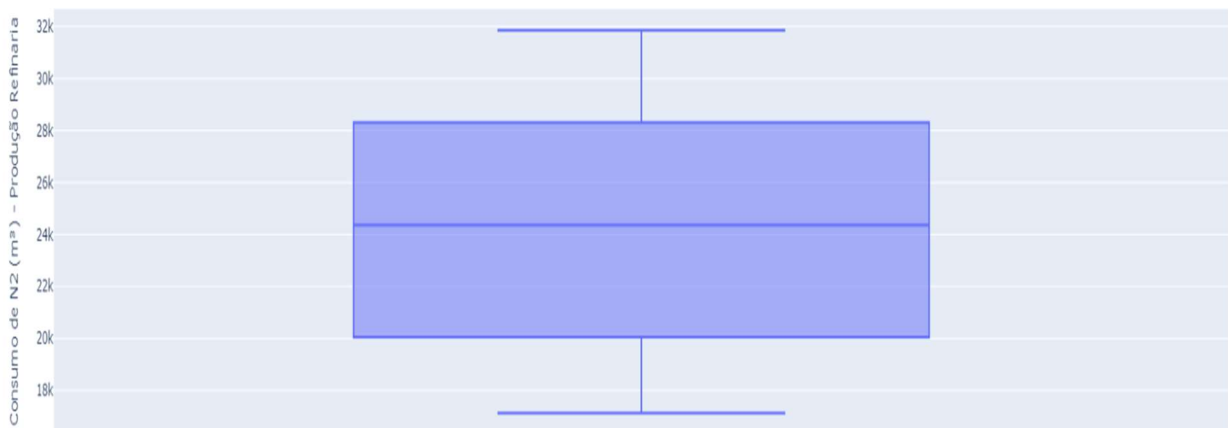
Fontes: Autor (2022)

Figura 41 – Histograma de Consumo de Gás.



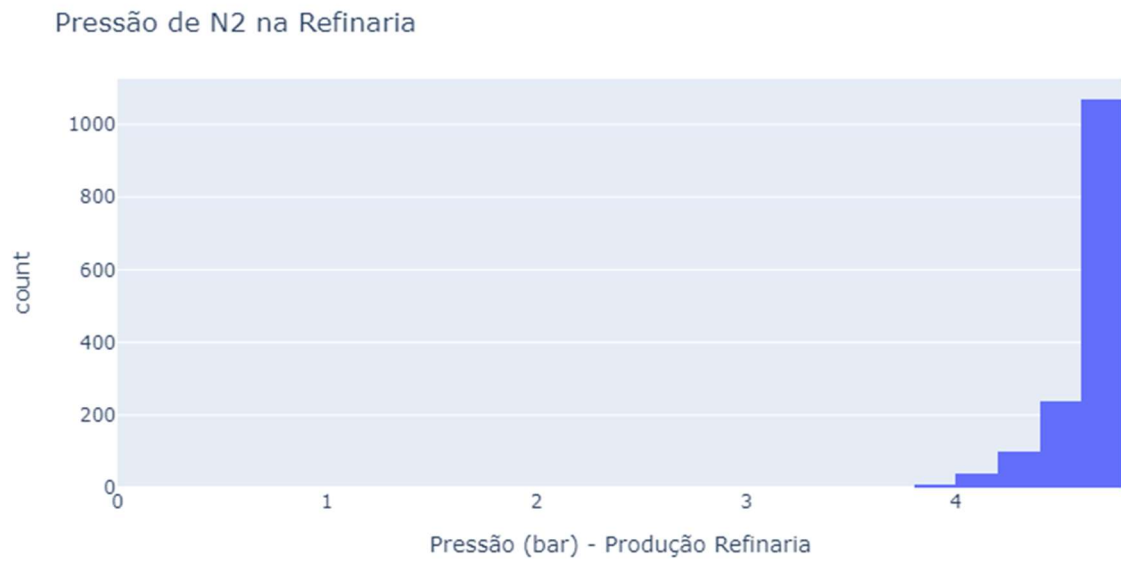
Fontes: Autor (2022)

Figura 42 – Box Plot de Consumo de Gás.



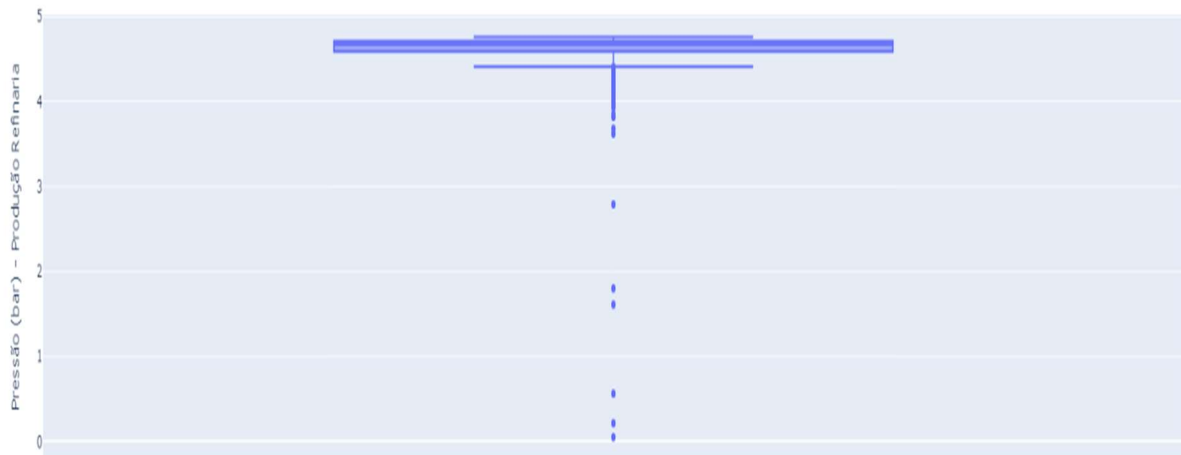
Fontes: Autor (2022)

Figura 43 – Histograma de Pressão de Gás.



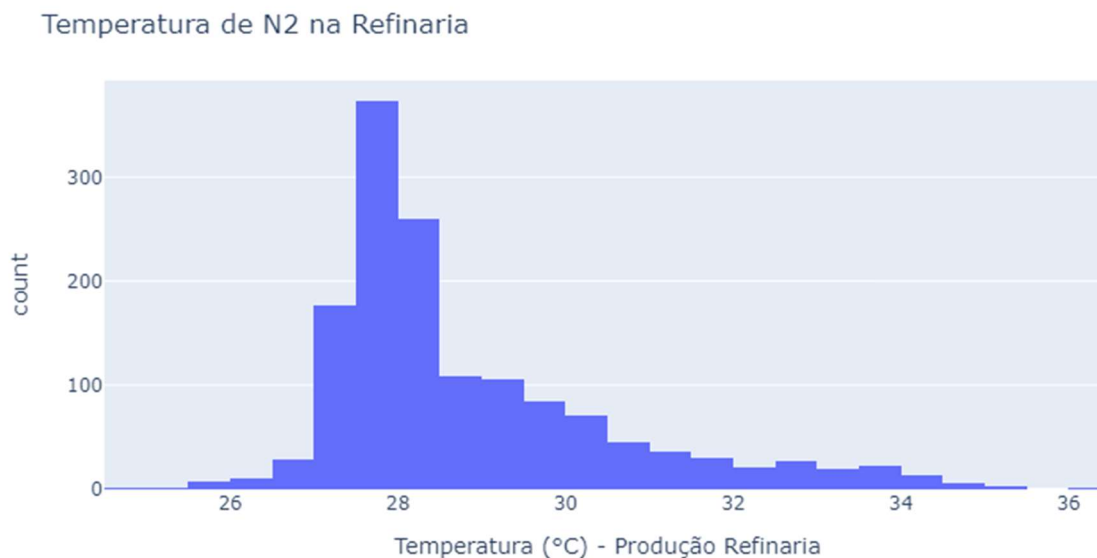
Fontes: Autor (2022)

Figura 44 – Box Plot de Pressão de Gás.



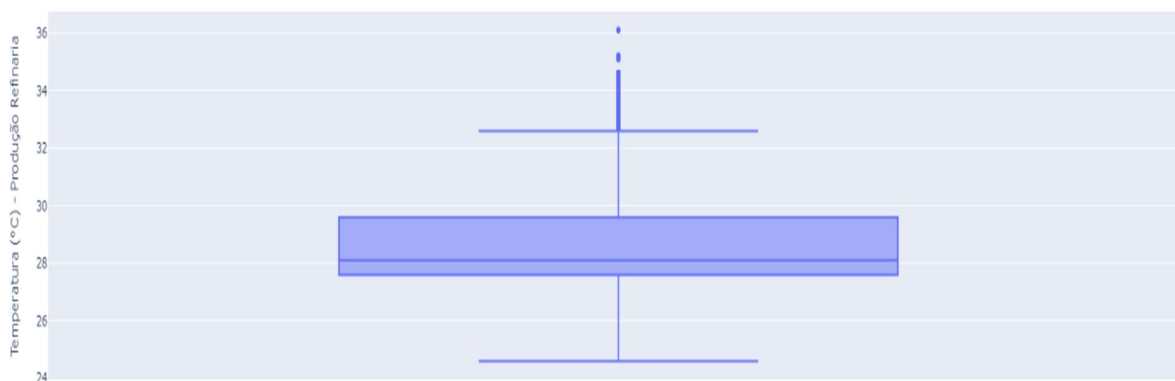
Fontes: Autor (2022)

Figura 45 – Histograma de Temperatura de Gás.



Fontes: Autor (2022)

Figura 46 – Box Plot de Temperatura de Gás.

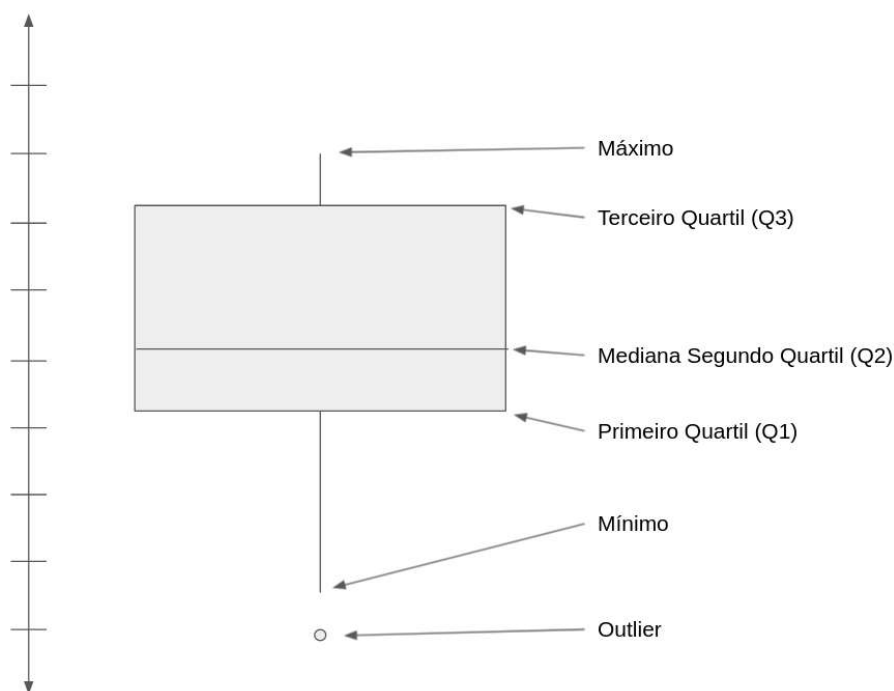


Fontes: Autor (2022)

Com as ilustrações dos diagramas de caixas mostrados anteriormente, será possível realizar uma segregação dos *Datas Frames*, de tal forma que exclua por completo a quantidade de valores fora do padrão.

Na Figura 47, está ilustrado os principais pontos presentes em um box plot.

Figura 47 – Descrição de Box Plots.



Fontes: Blog da Trybe (2022)

O box plot é composto por seis elementos: limite inferior, quartil 1, mediana (ou quartil 2), quartil 3, limite superior e outliers. Portanto, para interpretá-lo antes precisamos entender cada um desses elementos.

A mediana é o valor central de um conjunto de dados, quando esse conjunto está organizado de forma crescente, já os quartis, dividem esse conjunto em quatro (daí vem o nome quartis), cada um contendo 25% (um quarto) dos dados. Logo, percebe-se que o segundo quartil ocupará o mesmo lugar da mediana, enquanto que o primeiro quartil será a base da caixa e o terceiro quartil será o topo.

Além dos quartis, um box plot é composto pelos limites superior e inferior. Esses limites correspondem, respectivamente, ao maior e menor valor do conjunto de dados que não sejam outliers.

Equação 04 – Limites superior e inferior em um box plot.

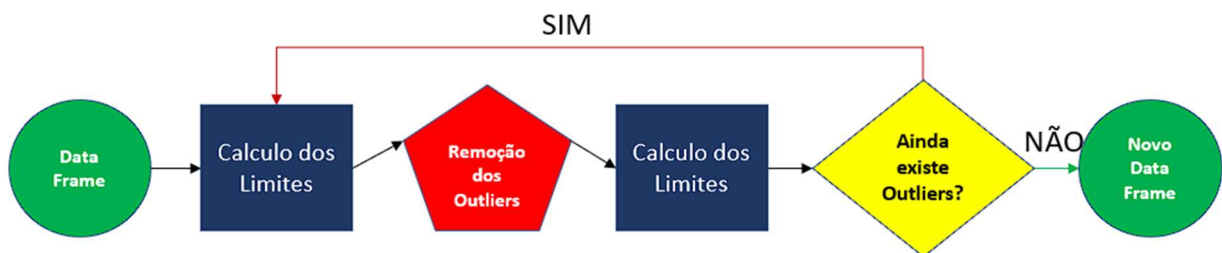
$$LIMs = Q3 + 1,5(Q3 - Q1)$$

$$LIMi = Q1 - 1,5(Q3 - Q1)$$

Portanto, utilizando-se das equações dos limites superiores e inferiores, poderemos realizar a retirada dos valores discrepantes através de uma simples associação.

Nas linhas de código da Figura 49 temos que, com a criação das formulas dos limites, aplicamos em uma função de loop, ilustrada na Figura 48, que calculará os valores superiores e inferiores constantemente, a mesma medida que qualquer valor acima ou abaixo deles, sejam excluídos das amostras.

Figura 48 – Representação esquemática da função loop.



Fontes: Autor (2022)

Figura 49 – Fórmulas de Tratamento de Outliers Aplicadas em Funções Loop.

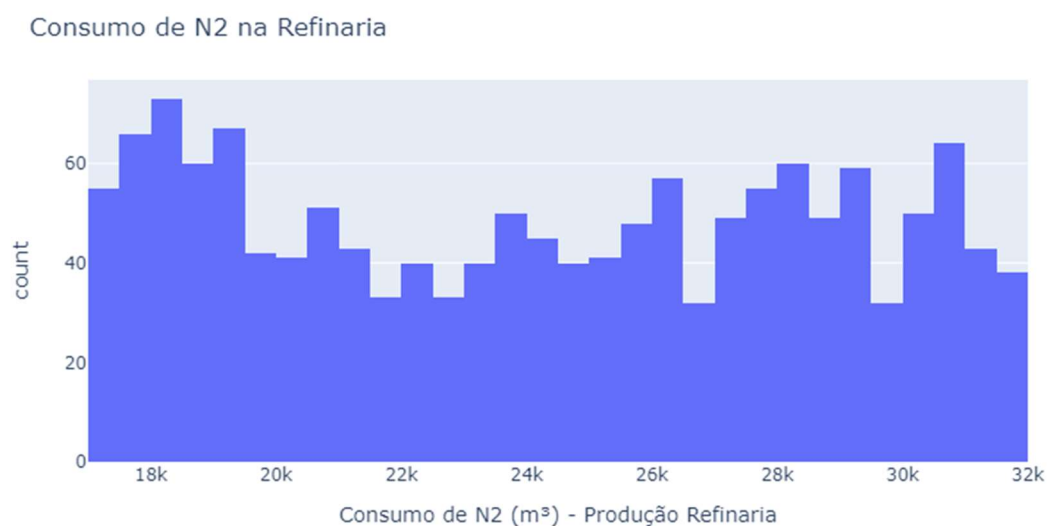
```

# Consumo
outlier_sup = df_cons['Consumo de N2 (m³) - Produção Refinaria'].quantile(q=0.75)
+ 1.5 * (df_cons['Consumo de N2 (m³) - Produção Refinaria'].quantile(q=0.75)
- df_cons['Consumo de N2 (m³) - Produção Refinaria'].quantile(q=0.25))
outlier_inf = df_cons['Consumo de N2 (m³) - Produção Refinaria'].quantile(q=0.25)
- 1.5 * (df_cons['Consumo de N2 (m³) - Produção Refinaria'].quantile(q=0.75)
- df_cons['Consumo de N2 (m³) - Produção Refinaria'].quantile(q=0.25))
while True:
    n=0
    df_cons = df_cons.loc[df_cons['Consumo de N2 (m³) - Produção Refinaria'] <= outlier_sup]
    df_cons = df_cons.loc[df_cons['Consumo de N2 (m³) - Produção Refinaria'] >= outlier_inf]
    outlier_sup = df_cons['Consumo de N2 (m³) - Produção Refinaria'].quantile(q=0.75)
    + 1.5 * (df_cons['Consumo de N2 (m³) - Produção Refinaria'].quantile(q=0.75)
    - df_cons['Consumo de N2 (m³) - Produção Refinaria'].quantile(q=0.25))
    outlier_inf = df_cons['Consumo de N2 (m³) - Produção Refinaria'].quantile(q=0.25)
    - 1.5 * (df_cons['Consumo de N2 (m³) - Produção Refinaria'].quantile(q=0.75)
    - df_cons['Consumo de N2 (m³) - Produção Refinaria'].quantile(q=0.25))
    for x in df_cons['Consumo de N2 (m³) - Produção Refinaria']:
        if x > outlier_sup or x < outlier_inf:
            n=1
    if n == 0:
        break
  
```

Fontes: Autor (2022)

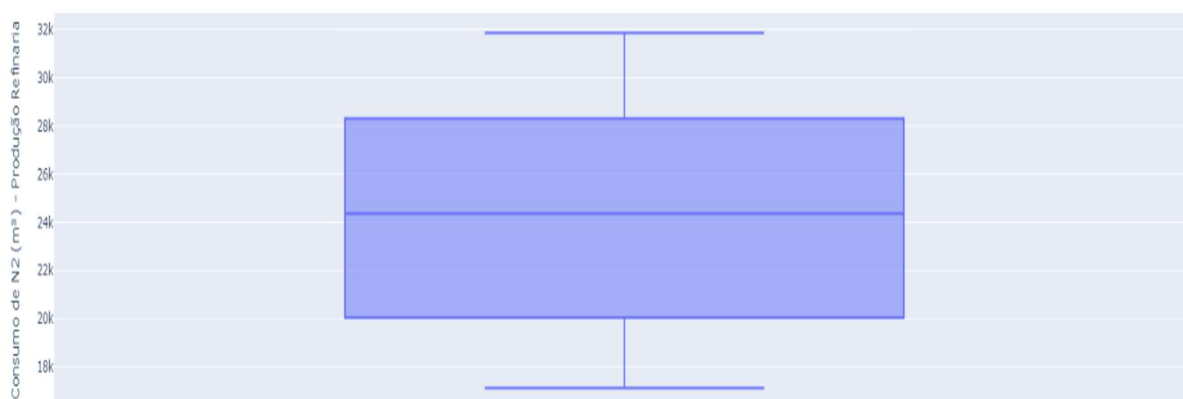
Com a aplicação da função loop para cada amostra, vemos que ao gerar as distribuições nos histogramas e box plots, nas Figuras 50, 51, 52, 53, 54 e 55, os valores apresentados pelos dados de pressão e temperatura de utilização de N₂, não possuem mais os outliers e seus histogramas mostram uma distribuição mais normalizada, em comparação aos valores antigos.

Figura 50 – Histograma de Consumo de Gás Tratado.



Fontes: Autor (2022)

Figura 51 – Box Plot de Consumo de Gás Tratado.



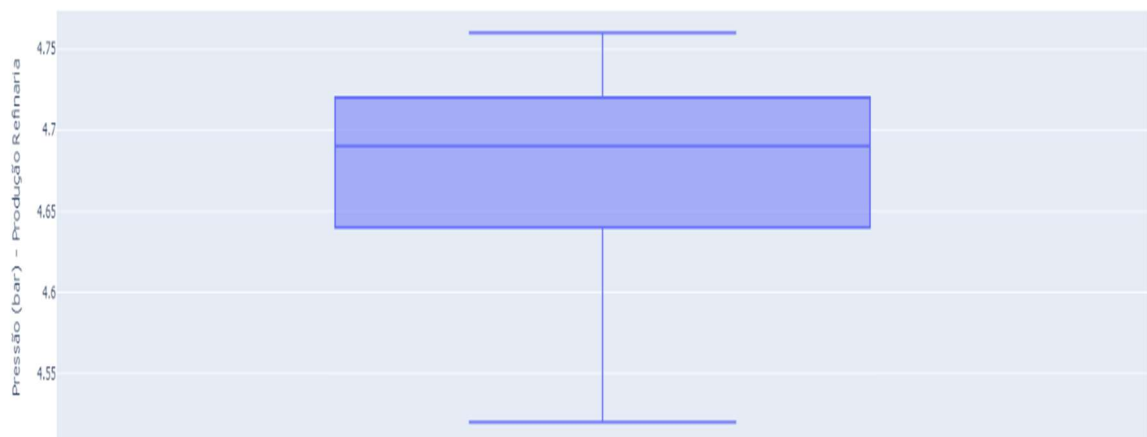
Fontes: Autor (2022)

Figura 52 – Histograma de Pressão de Gás Tratado.



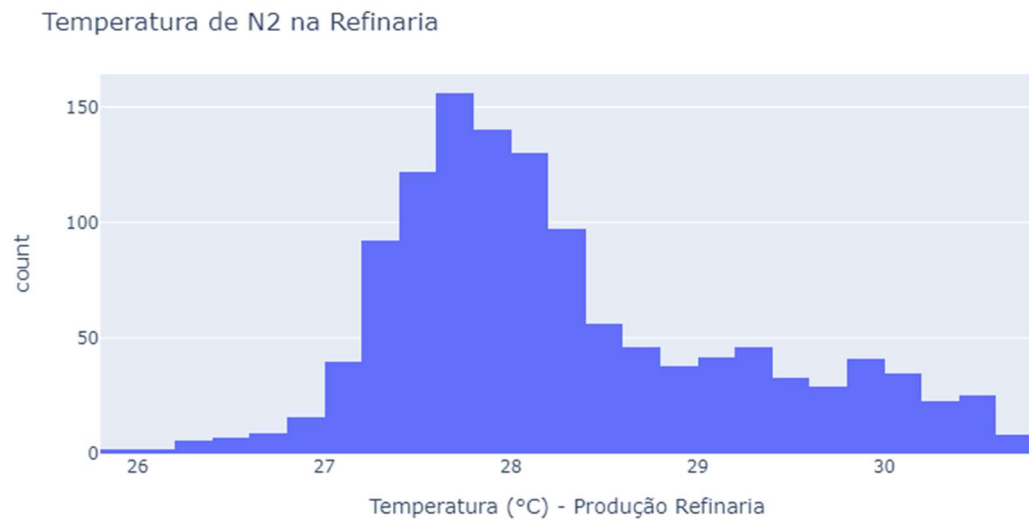
Fontes: Autor (2022)

Figura 53 – Box Plot de Pressão de Gás Tratado.



Fontes: Autor (2022)

Figura 54 – Histograma de Temperatura de Gás Tratado.



Fontes: Autor (2022)

Figura 55 – Box Plot de Temperatura de Gás Tratado.



Fontes: Autor (2022)

Para finalizar esta etapa, foi aplicada, apenas, uma função responsável por resetar os indices dos *Data Frames*, apresentado na Figura 56, pois como os mesmos passaram por um processo de exclusão de valores, seria normal que algumas linhas permanecessem com valores nulos, o que não seria ideal para as posteriores etapas do projeto, logo, essa linha de código tratou de reorganizar as tabelas, excluindo os valores nulos.

Figura 56 – Função de resetar indicie.

```
df_cons = df_cons.reset_index(drop=True)
df_pre = df_pre.reset_index(drop=True)
df_temp = df_temp.reset_index(drop=True)
```

Fontes: Autor (2022)

3.4.5 *Divisão das Amostras*

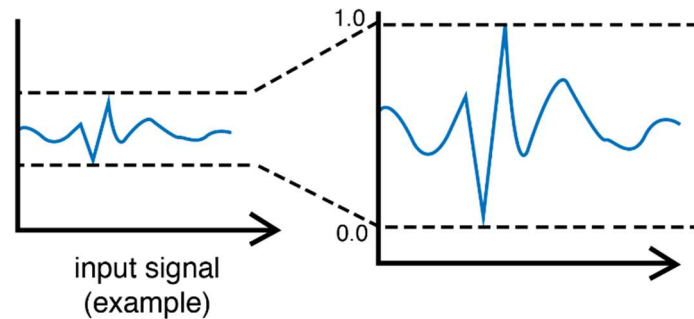
Quando um conjunto de dados tem valores de colunas diferentes em escalas drasticamente diferentes, fica difícil analisar as tendências e padrões e comparar os recursos ou colunas. Assim, nos casos em que todas as colunas tenham uma diferença significativa em suas escalas, devem ser modificadas de forma que todos aqueles valores caiam na mesma escala, este processo é denominado *Escalonamento*.

Resumindo, o escalonamento, ou dimensionamento, dos dados é um artifício estatístico utilizado em diversos processos de regressão para evitar que grandes diferenças de valores venham a afetar tendenciosamente a exatidão e precisão do modelo de regressão. Este método por sua vez, utiliza-se da equação a seguir, para que todos os valores se encontrem em um intervalo de 0 a 1, como ilustrado na Figura 57.

Equação 05 – Formula do escalonamento de um conjunto de dados.

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

Figura 57 – Representação Gráfica do Dimensionamento dos Dados.



Fontes: Github (2022)

Na interface Python o dimensionamento é feito através da função “MinMaxScaler()” e está apresentado nos códigos da Figura 58.

Figura 58 – Função de Normalização dos Dados.

```
##### Etapa 1 - Selecione os dados para modelagem e aplique o dimensionamento MinMax
# Consumo
X_cons=df_cons[['Consumo de N2 (m³) - Produção Refinaria']]
scaler = MinMaxScaler()
X_cons_scaled=scaler.fit_transform(X_cons)

# Pressão
X_pre=df_pre[['Pressão (bar) - Produção Refinaria']]
scaler = MinMaxScaler()
X_pre_scaled=scaler.fit_transform(X_pre)

# Temperatura
X_temp=df_temp[['Temperatura (°C) - Produção Refinaria']]
scaler = MinMaxScaler()
X_temp_scaled=scaler.fit_transform(X_temp)
```

Fontes: Autor (2022)

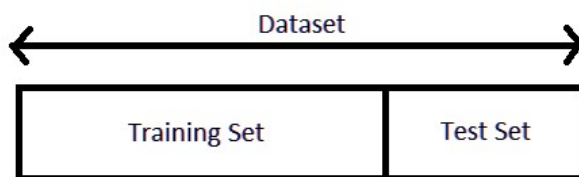
Antes de prosseguir para a criação do modelo de rede neural, antes deveremos realizar mais uma etapa de tratamento dos dados, que é a divisão do conjunto de dados, ou *Dataset*, e a criação dos conjuntos de treinamento e de teste.

O princípio da divisão do *Dataset* é bem simples, basicamente será coletada uma parcela dos dados em estudo, para cada um dos 3 *Data Frames*, onde esta parcela será utilizada para criar outro *Dataset*, chamado de *Dataset de Teste*, e o restante do conjunto será chamado de *Dataset de Treinamento*. Geralmente essa parcela representa de 10% a 40% da Tabela de dados, no caso do estudo em foco, foram utilizados os valores de 35% para a Tabela de *Consumo*, 40% para a Tabela de *Pressão* e 40% para a Tabela de *Temperatura*. Estes valores foram escolhidos através de diversos testes, onde foi escolhido o valor que fornecesse o modelo com menor erro.

Vale ressaltar que a coleta de valores é realizada de forma aleatória, ou seja, os valores coletados não necessariamente são sequências, isso é importante para evitar certas influências sazonais na construção do modelo.

Como o nome já diz, o *Dataset* de Treinamento será utilizado para realizar o treinamento do modelo de rede neural, será através de seus dados que o modelo realizará as operações computacionais que permitirão a criação de uma função que possa ser usada para descrever o comportamento do sistema. Com o modelo já estabelecido, o mesmo terá seus resultados comparados com o *Dataset* de Teste, onde poderemos avaliar se os valores são aceitáveis ou se o modelo deverá ser refeito, como mostra a Figura 59.

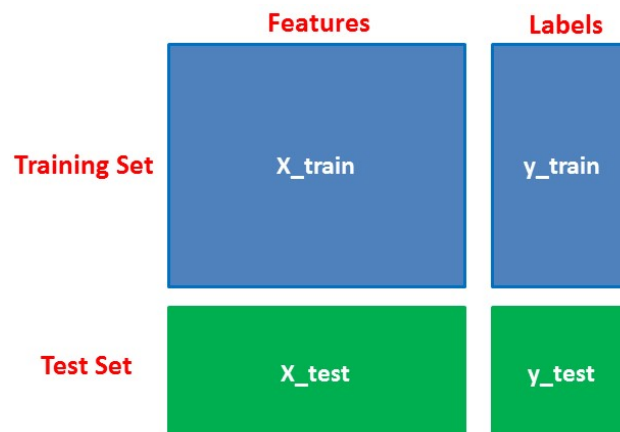
Figura 59 – Divisão do Conjunto de Dados.



Fontes: G.Rinu (2019)

Para finalizar as etapas de tratamento de dados, iremos dividir os conjuntos de dados de treinamento e teste em mais duas categorias. De modo geral, criaremos um *Dataset* que receberá a todos os valores que servirão como como variáveis independentes (X) para que seja predito o valor de um outro *Dataset* que conterà as variáveis dependentes (Y). A Figura 60 ilustra melhor como ocorre a divisão.

Figura 60 – Separação das variáveis a serem preditas.

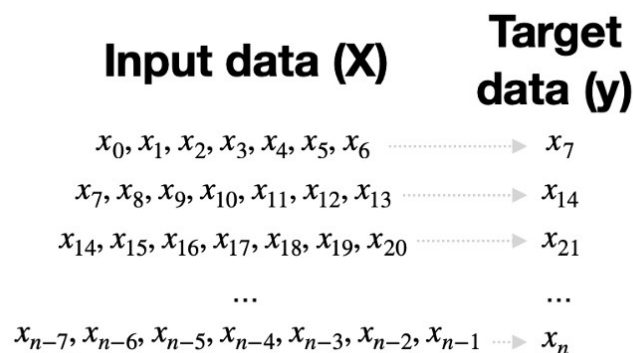


Fontes: Github (2022)

Como os valores de consumo, pressão e temperatura não possuem uma alta correlação, a maneira mais adequada de se criar os conjuntos dependentes e independentes será utilizando os valores das próprias variáveis principais. Para exemplificar, temos que para o conjunto de dados de treinamento para o consumo de gás da refinaria, a predição de um valor da amostra, será feita utilizando uma quantidade pré-determinada (*time step*) de valores anteriores da mesma amostra.

Na Figura 61 abaixo, temos um exemplo em que o modelo de predição da variável dependente “y” foi criado, utilizando os 8 valores anteriores como variáveis independentes “X”.

Figura 61 – Representação do *Dataset* de Treinamento.



Fontes: D.Saul (2022)

Com a teoria já desenvolvida, a aplicação no código fonte se deu primeiramente com a criação de uma função “*prep_data()*” que seria utilizada posteriormente para a formação das variáveis dependentes e independentes, como mostra abaixo.

No entanto, antes da aplicação da função mencionada acima, foram formados os conjuntos de dados de treinamento e de teste, pela função “*train_test_split()*”. Percebe-se, nas Figuras 62 e 63, que os *times steps* usados foram 1 para o consumo, 3 para a pressão e 6 para a temperatura.

Figura 62 – Função de Preparação dos Dados.

```
##### Etapa 0 - Usaremos esta função na etapa 3 para obter os dados na forma correta
def prep_data(datain, time_step):
    # 1. Vetor Y
    # Primeiro, crie um vetor com índices para y elementos com base no time_step escolhido
    y_indices = np.arange(start=time_step, stop=len(datain), step=time_step)
    # Crie uma matriz y com base nos índices acima
    y_tmp = datain[y_indices]

    # 2. Vetor X
    # Queremos ter o mesmo número de linhas para X que temos para y
    rows_X = len(y_tmp)
    # Como o último elemento em y_tmp pode não ser o último elemento do datain, vamos garantir que a matriz X pare com o último y
    X_tmp = datain[range(time_step*rows_X)]
    # Agora pegue esta matriz e remodele-a na forma desejada
    X_tmp = np.reshape(X_tmp, (rows_X, time_step, 1))
    return X_tmp, y_tmp
```

Fontes: Autor (2022)

Figura 63 – Criação das amostras de treino e teste.

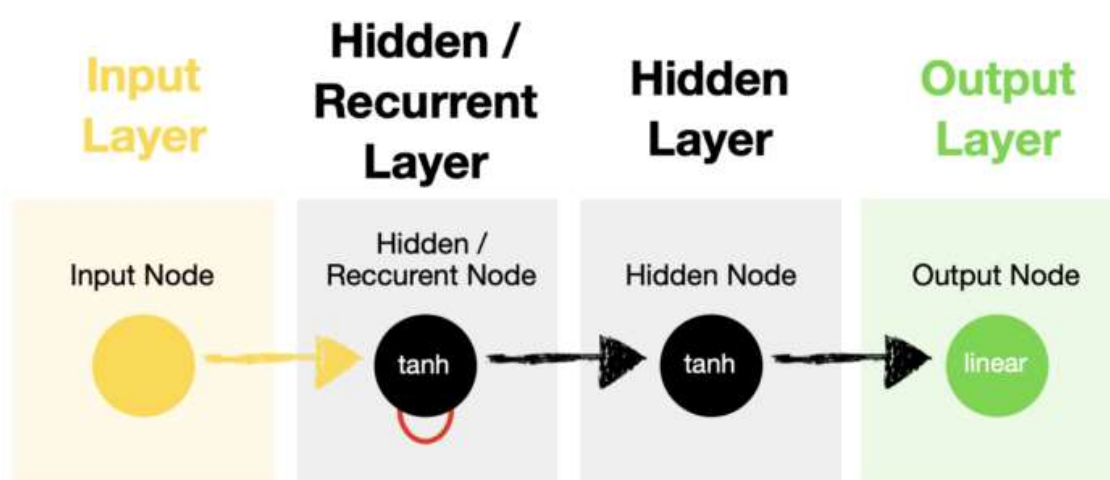
```
##### Etapa 2 - Criar amostras de treinamento e teste
# Consumo
train_data_cons, test_data_cons = train_test_split(X_cons_scaled, test_size=0.35, shuffle=False)
time_step_cons = 1
X_train_cons, y_train_cons = prep_data(train_data_cons, time_step_cons)
X_test_cons, y_test_cons = prep_data(test_data_cons, time_step_cons)
# Pressão
train_data_pre, test_data_pre = train_test_split(X_pre_scaled, test_size=0.4, shuffle=False)
time_step_pre = 3
X_train_pre, y_train_pre = prep_data(train_data_pre, time_step_pre)
X_test_pre, y_test_pre = prep_data(test_data_pre, time_step_pre)
# Temperatura
train_data_temp, test_data_temp = train_test_split(X_temp_scaled, test_size=0.4, shuffle=False)
time_step_temp = 6
X_train_temp, y_train_temp = prep_data(train_data_temp, time_step_temp)
X_test_temp, y_test_temp = prep_data(test_data_temp, time_step_temp)
```

Fontes: Autor (2022)

3.4.6 Criação do Modelo

Finalmente, após a execução de todas as etapas de tratamento dos dados, a próxima fase do projeto será utilizar os dados de treinamento para criar um modelo de rede neural. No início do trabalho, foram apresentados diversos tipos diferentes de redes neurais, no entanto para esse estudo, foi selecionado apenas o método de *Redes Neurais Recorrentes* (RNN), justamente por se adequar melhor ao estudo de variáveis no formato de séries temporais. A forma das camadas de RNN podem ser visualizadas na Figura 64.

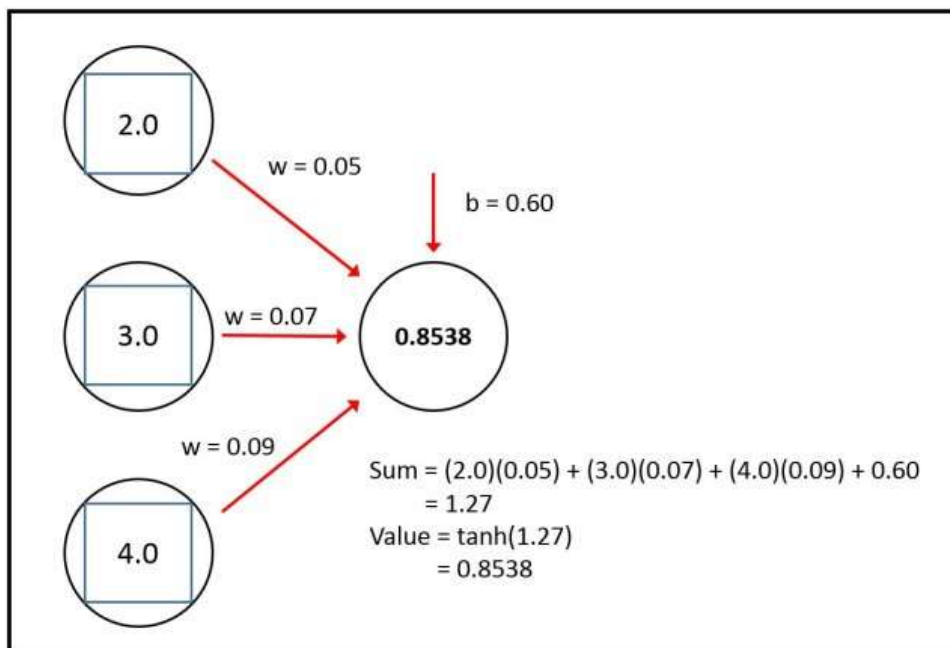
Figura 64 – Representação das camadas da RNN.



Fontes: D.Saul (2022)

No processo de criação de uma RNN, utilizaremos os conjuntos de variáveis independentes para obter um valor próximo o suficiente da variável dependente, para isso, as variáveis de entrada (*Input Layer*) se conectaram, através de pesos (W) e constantes (b) arbitrárias, com um número pré-determinado de nós (*Nodes*), localizados nas camadas ocultas (*Hidden Layers*) da rede neural. Estas camadas ocultas por sua vez receberão valores distintos, por meio das funções de ativação selecionadas. A Figura 65 ilustra um exemplo de conexão, em que a função de ativação escolhida foi a *tangente hiperbólica*.

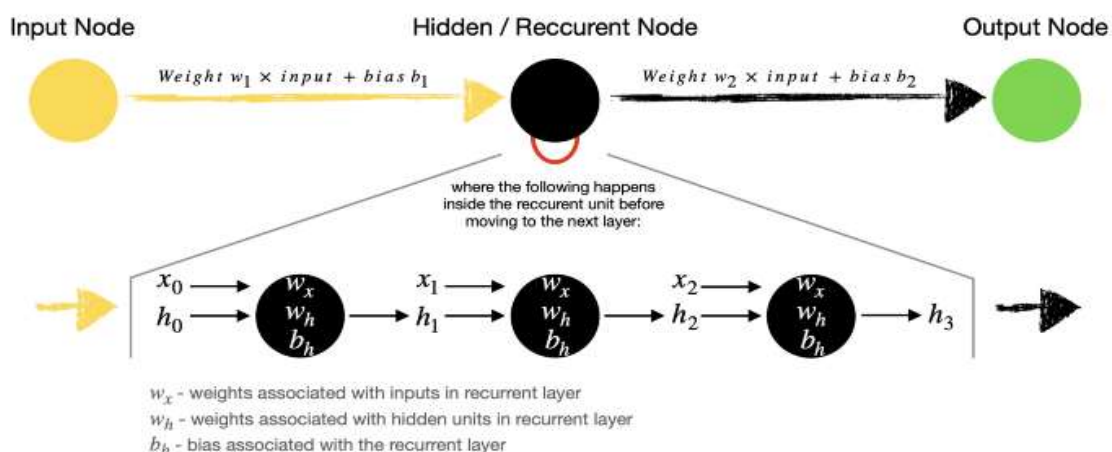
Figura 65 – Ilustração das conexões existentes entre os nós de uma rede neural.



Fontes: D.Saul (2022)

Para o caso de uma rede neural feed forward típica, o nó oculto teria dois parâmetros (peso e constante), no entanto, uma camada recorrente tem três parâmetros para otimizar: peso para a entrada, peso para a unidade oculta e constante. Veja na Figura 66, uma representação de uma rede neural recorrente com variáveis de entradas categorizadas em 3 *time-steps*.

Figura 66 – Ilustração das operações realizadas nas camadas recorrentes.



Fontes: towardsdatascience (2022)

Para a melhor construção do modelo, devemos antes esclarecer alguns pontos fundamentais, referentes ao formato do mesmo. Primeiramente, serão utilizados 4 nós para cada camada oculta.

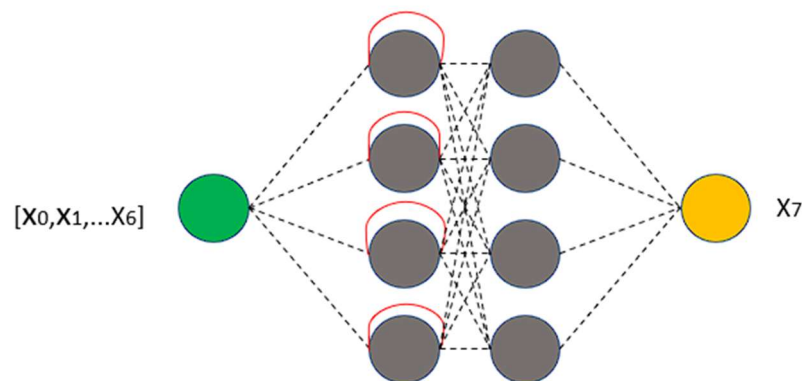
Em segundo lugar, as funções de ativação respectivas para cada camadas ocultas estarão distribuídas da seguinte forma:

- Para o modelo de consumo de gás, todas as funções de ativação serão lineares.
- Para os modelos de pressão e temperatura de gás, as camadas ocultas terão funções tangentes hiperbólicas como funções de ativação, porém a camada de saída terá uma função linear para tal.

As funções de ativações foram escolhidas com base nos resultados obtidos em repetidas operações, o conjunto de funções de ativação que fornecessem modelos com menores erros seriam escolhidos.

Adiante, teremos, na Figura 67, uma representação gráfica da nossa rede neural, seguida pelo código fonte contendo os modelos para cada conjunto de dados de treinamento, apresentada na Figura 68.

Figura 67 – Diagrama do Modelo da RNN Utilizada.



Fontes: Autor (2022)

Figura 68 – Código da Estrutura da Rede Neural.

```

##### Passo 4 - Especifique a estrutura de uma Rede Neural
# Consumo
model_cons = Sequential(name="First-RNN-Model") # Modelo
model_cons.add(Input(shape=(time_step_cons,1),
                    name='Input-Layer')) # Camada de entrada - precisa especificar a forma das entradas
model_cons.add(SimpleRNN(units=4,
                        activation='linear',
                        name='Hidden-Recurrent-Layer')) # Camada recorrente oculta, Linear(x) = x
model_cons.add(Dense(units=4,
                    activation='linear',
                    name='Hidden-Layer')) # Camada Oculta, Linear(x) = x
model_cons.add(Dense(units=1,
                    activation='linear',
                    name='Output-Layer')) # Camada de Saida, Linear(x) = x

# Pressão
model_pre = Sequential(name="First-RNN-Model") # Modelo
model_pre.add(Input(shape=(time_step_pre,1),
                    name='Input-Layer')) # Camada de entrada - precisa especificar a forma das entradas
model_pre.add(SimpleRNN(units=4,
                        activation='tanh',
                        name='Hidden-Recurrent-Layer')) # Camada recorrente oculta, Tanh(x) = sinh(x)/cosh(x) = ((exp(x) - exp(-x))/(exp(x) + exp(-x)))
model_pre.add(Dense(units=4,
                    activation='tanh',
                    name='Hidden-Layer')) # Camada Oculta, Tanh(x) = sinh(x)/cosh(x) = ((exp(x) - exp(-x))/(exp(x) + exp(-x)))
model_pre.add(Dense(units=1,
                    activation='linear',
                    name='Output-Layer')) # Camada de Saida, Linear(x) = x

# Temperatura
model_temp = Sequential(name="First-RNN-Model") # Modelo
model_temp.add(Input(shape=(time_step_temp,1),
                    name='Input-Layer')) # Camada de entrada - precisa especificar a forma das entradas
model_temp.add(SimpleRNN(units=4,
                        activation='tanh',
                        name='Hidden-Recurrent-Layer')) # Camada recorrente oculta, Tanh(x) = sinh(x)/cosh(x) = ((exp(x) - exp(-x))/(exp(x) + exp(-x)))
model_temp.add(Dense(units=4,
                    activation='tanh',
                    name='Hidden-Layer')) # Camada Oculta, Tanh(x) = sinh(x)/cosh(x) = ((exp(x) - exp(-x))/(exp(x) + exp(-x)))
model_temp.add(Dense(units=1,
                    activation='linear',
                    name='Output-Layer')) # Camada de Saida, Linear(x) = x

```

Fontes: Autor (2022)

Logo após a formulação dos modelos, foi definido, como na Figura 69, os métodos a serem utilizados para a medição dos erros dos mesmos, que serão melhor tratados mais adiante.

Figura 69 – Definição das medidas avaliativas.

```
##### Passo 5 - Compile keras model
# Consumo
model_cons.compile(optimizer='adam',
                   loss='mean_squared_error',
                   metrics=['MeanSquaredError', 'MeanAbsoluteError'],
                   loss_weights=None,
                   weighted_metrics=None,
                   run_eagerly=None,
                   steps_per_execution=1
                   )
# Pressão
model_pre.compile(optimizer='adam',
                  loss='mean_squared_error',
                  metrics=['MeanSquaredError', 'MeanAbsoluteError'],
                  loss_weights=None,
                  weighted_metrics=None,
                  run_eagerly=None,
                  steps_per_execution=1
                  )
# Temperatura
model_temp.compile(optimizer='adam',
                   loss='mean_squared_error',
                   metrics=['MeanSquaredError', 'MeanAbsoluteError'],
                   loss_weights=None,
                   weighted_metrics=None,
                   run_eagerly=None,
                   steps_per_execution=1
                   )
)
```

Fontes: Autor (2022)

Por fim, o modelo é então aplicado a uma função de execução, como na Figura 70, onde serão compilados, durante 100 vezes, todos os dados referentes aos conjuntos de treinamento, com o intuito de proporcionar uma maior taxa de aprendizado da rede neural, tendo em vista que à medida que o modelo é executado, todo o sistema identifica cada padrão de comportamento com maior exatidão. A quantidade de épocas, no entanto, possui como principal limitador a capacidade de processamento do computador em que o sistema está incluído, limitando assim o desempenho do sistema. As épocas podem ser visualizadas na Figura 71.

Figura 70 – Definição das configurações da RN.

```
##### Etapa 6 - Ajuste o modelo keras no conjunto de dados
model_cons.fit(X_train_cons,
               y_train_cons,
               batch_size=1,
               epochs=100,
               verbose='auto',
               callbacks=None,
               validation_split=0.0,
               #validation_data=(X_test, y_test),
               shuffle=True,
               class_weight=None,
               sample_weight=None,
               initial_epoch=0,
               steps_per_epoch=None,
               validation_steps=None,
               validation_batch_size=None,
               validation_freq=1,
               max_queue_size=10,
               workers=1, # Predefinição=1,
               use_multiprocessing=False,
               )
```

Fontes: Autor (2022)

Figura 71 – Épocas dos processos.

```
Epoch 1/100
945/945 [=====] - 1s 942us/step - loss: 0.0071 - mean_squared_error: 0.0071 - mean_absolute_error: 0.0446
Epoch 2/100
945/945 [=====] - 1s 954us/step - loss: 8.6834e-07 - mean_squared_error: 8.6834e-07 - mean_absolute_error: 6.6383e-06
Epoch 3/100
945/945 [=====] - 1s 961us/step - loss: 9.7932e-07 - mean_squared_error: 9.7932e-07 - mean_absolute_error: 7.1378e-06
Epoch 4/100
945/945 [=====] - 1s 938us/step - loss: 1.1356e-06 - mean_squared_error: 1.1356e-06 - mean_absolute_error: 7.8621e-06
Epoch 5/100
945/945 [=====] - 1s 972us/step - loss: 1.2522e-06 - mean_squared_error: 1.2522e-06 - mean_absolute_error: 8.0859e-06
Epoch 6/100
945/945 [=====] - 1s 963us/step - loss: 1.7480e-06 - mean_squared_error: 1.7480e-06 - mean_absolute_error: 9.8170e-06
Epoch 7/100
945/945 [=====] - 1s 1ms/step - loss: 2.0925e-06 - mean_squared_error: 2.0925e-06 - mean_absolute_error: 0.0011
Epoch 8/100
945/945 [=====] - 1s 935us/step - loss: 3.3791e-06 - mean_squared_error: 3.3791e-06 - mean_absolute_error: 0.0014
Epoch 9/100
945/945 [=====] - 1s 915us/step - loss: 5.0517e-06 - mean_squared_error: 5.0517e-06 - mean_absolute_error: 0.0016
Epoch 10/100
945/945 [=====] - 1s 925us/step - loss: 5.0170e-06 - mean_squared_error: 5.0170e-06 - mean_absolute_error: 0.0016
Epoch 11/100
945/945 [=====] - 1s 927us/step - loss: 5.7632e-06 - mean_squared_error: 5.7632e-06 - mean_absolute_error: 0.0018
Epoch 12/100
945/945 [=====] - 1s 918us/step - loss: 3.4852e-06 - mean_squared_error: 3.4852e-06 - mean_absolute_error: 0.0014
Epoch 13/100
...
Epoch 99/100
945/945 [=====] - 1s 970us/step - loss: 3.3970e-06 - mean_squared_error: 3.3970e-06 - mean_absolute_error: 0.0014
Epoch 100/100
945/945 [=====] - 1s 981us/step - loss: 3.8783e-06 - mean_squared_error: 3.8783e-06 - mean_absolute_error: 0.0015
```

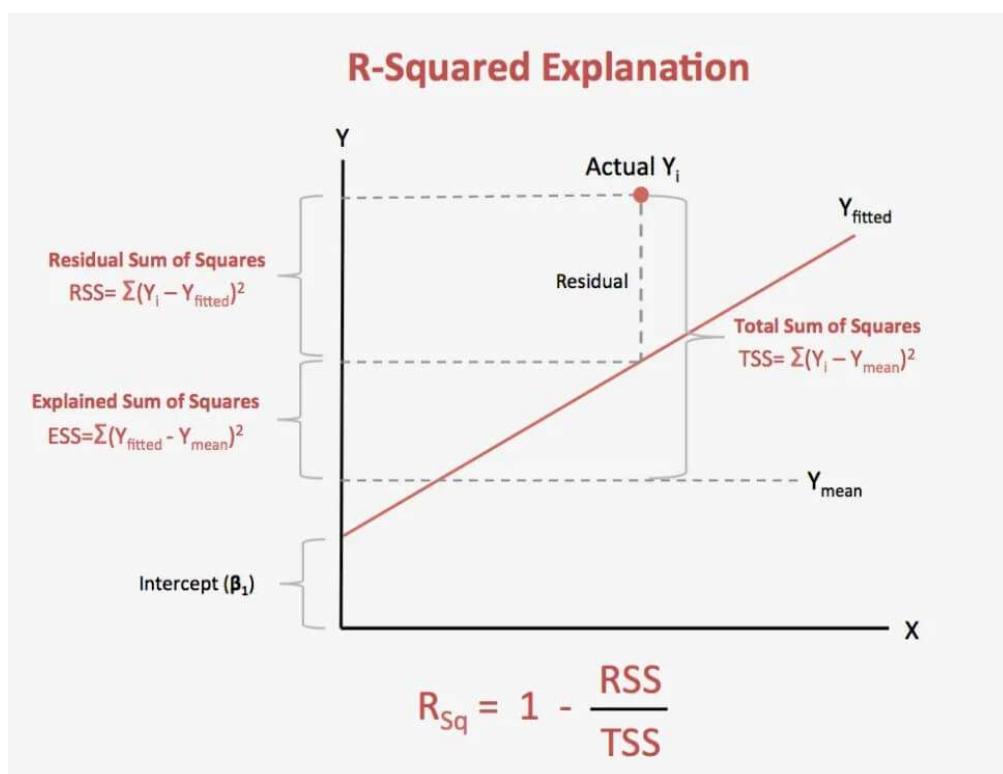
Fontes: Autor (2022)

3.4.7 Medidas Avaliativas

Com os modelos finalizados, chega-se a hora de avaliar como está se comportando os valores criados pela metodologia RNN, para isso, a ação tomada foi a de plotar os valores preditos juntamente com os valores do *Dataset* de teste, e submeter os dados preditos a um grupo de equações e fundamentos estatísticos que os comparam aos dados de treino e teste.

Dentre os métodos avaliativos utilizados temos que o principal e de maior aplicação no que tange os estudos voltados para a área de machine learning, é o “R-Squared” (R quadrado ou R^2). O R-quadrado é uma medida estatística de quão próximos os dados estão da linha de regressão ajustada, como mostra a Figura 72. O R^2 varia entre 0 e 1, por vezes sendo expresso em termos percentuais. Nesse caso, expressa a quantidade da variância dos dados que é explicada pelo modelo linear, assim, quanto maior o R^2 , mais explicativo é o modelo linear, ou seja, melhor ele se ajusta à amostra.

Figura 72 – Representação Gráfica do R^2 .



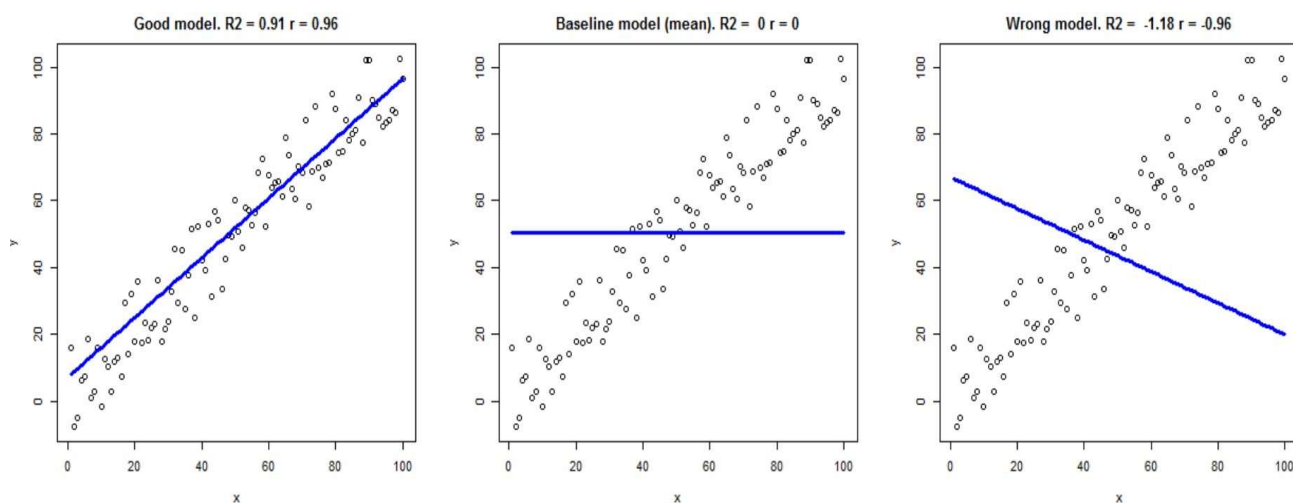
Fontes: Forex Academy(2021)

Equação 06 – Equação do R^2 .

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

Na Figura 73, podemos conferir representações gráficas que nos ajudam a entender como o valor do R^2 está relacionado com o comportamento e eficácia do modelo de regressão avaliado.

Figura 73 – Categorias de R^2 .



Fontes: Statistics How To (2022)

Logo, com as informações acerca dos métodos avaliativos e com os modelos criados, poderemos então criar conjuntos de dados de predição e compará-los com os valores reais dos comportamentos das series temporais.

Por meio da interface python, aplicou-se a função “*predict()*” associada aos dados de treinamento e teste, para que fossem produzidos suas análises preditivas. Veja na Figura 74:

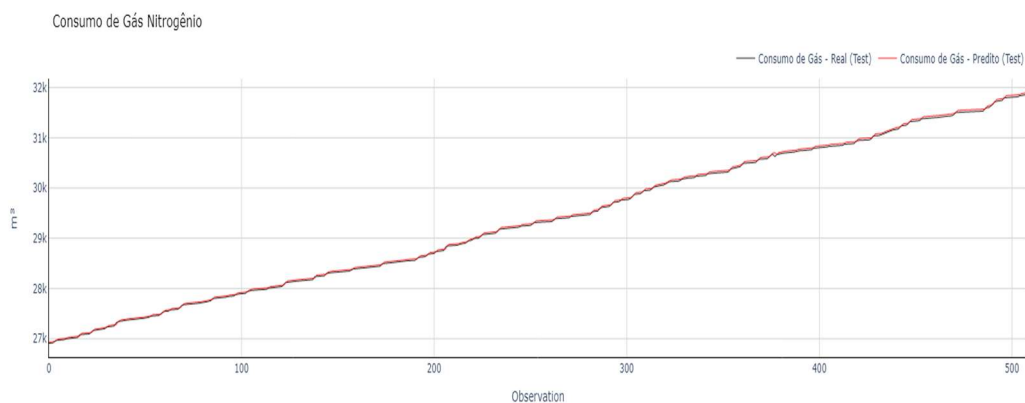
Figura 74 – Criação dos *Datasets* de Predição, e Função R^2 para cada Variável.

```
##### Etapa 7 - Use o modelo para fazer previsões
print('=====Consumo=====')
# Prever o resultado nos dados de treinamento
pred_train_cons = model_cons.predict(X_train_cons)
# Prever o resultado nos dados de teste
pred_test_cons = model_cons.predict(X_test_cons)
print('R²: ', r2_score(y_test_cons,pred_test_cons)*100,"%")
print('=====Pressão=====')
# Prever o resultado nos dados de treinamento
pred_train_pre = model_pre.predict(X_train_pre)
# Prever o resultado nos dados de teste
pred_test_pre = model_pre.predict(X_test_pre)
print('R²: ', r2_score(y_test_pre,pred_test_pre)*100,"%")
print('=====Temperatura=====')
# Prever o resultado nos dados de treinamento
pred_train_temp = model_temp.predict(X_train_temp)
# Prever o resultado nos dados de teste
pred_test_temp = model_temp.predict(X_test_temp)
print('R²: ', r2_score(y_test_temp,pred_test_temp)*100,"%")
```

Fontes: Autor (2022)

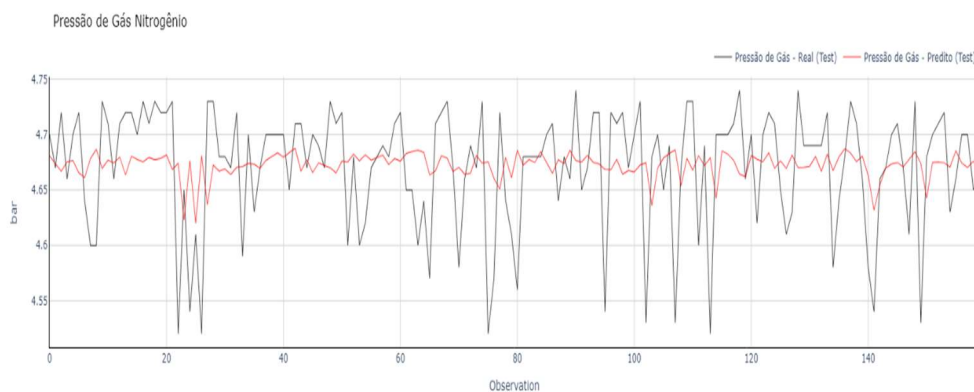
Mais a diante, nas Figuras 75, 76 e 77, podemos ter uma noção de como o modelo preditivo está se comportando, quando colocamos os mesmos juntamente com os valores reais. É notório que o modelo se comportou de forma quase que perfeita quando aplicado aos dados de consumo de N_2 , ao mesmo passo que também se apresenta de forma satisfatória para a temperatura do gás. No entanto, os valores resultantes da predição de pressão de nitrogênio aparentam uma não conformidade com os dados reais, o que deverá ser avaliado com cautela.

Figura 75 – Modelo aplicado aos dados de teste de consumo.



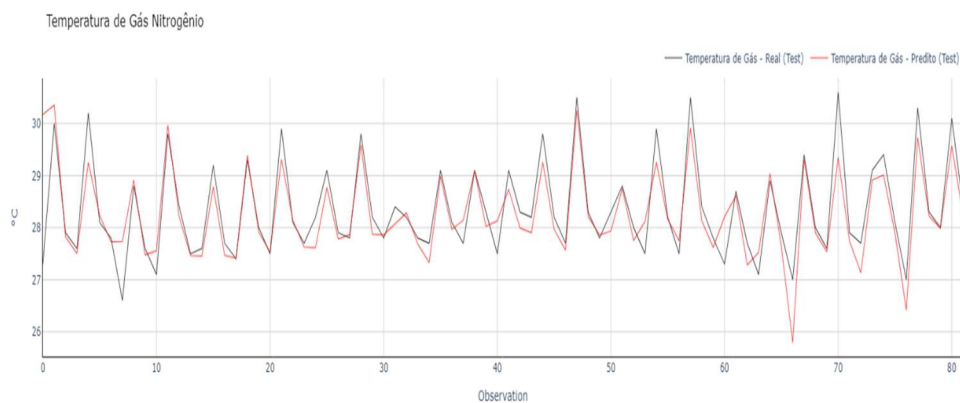
Fontes: Autor (2022)

Figura 76 – Modelo aplicado aos dados de teste de pressão.



Fontes: Autor (2022)

Figura 77 – Modelo aplicado aos dados de teste de temperatura.



Fontes: Autor (2022)

Como um processo avaliativo adicional, os valores de regressão também foram submetidos a uma definição do valor referente aos erros quadráticos médios dos dados, que fornece a média de diferença quadrática entre a predição do modelo e o valor de destino, tendo como valor ideal o número 0, como indica a Equação 07 e a Figura 78:

Equação 07 – Equação do Erro Quadrático Médio.

$$MSE = \frac{1}{n} \sum \left(\underbrace{y - \hat{y}}_{\substack{\text{The square of the difference} \\ \text{between actual and} \\ \text{predicted}}} \right)^2$$

Figura 78 – Função para mostrar o resumo geral do modelo.

```
##### Etapa 8 - Resumo do desempenho do modelo
print("")
print('----- Resumo do modelo - Consumo -----')
model_cons.summary() # imprimir resumo do modelo
print("")
print('----- Pesos e bias -----')
print("Observe que o último parâmetro em cada camada é bias, enquanto o restante são pesos")
print("")
for layer in model_cons.layers:
    print(layer.name)
    for item in layer.get_weights():
        print(" ", item)
print("")
print('----- Avaliação dos dados de treinamento -----')
print("MSE: ", mean_squared_error(y_train_cons, pred_train_cons))
print("")

print('----- Avaliação em dados de teste -----')
print("MSE: ", mean_squared_error(y_test_cons, pred_test_cons))
print("")
```

Fontes: Autor (2022)

3.4.8 *Análise Preditiva*

Nesta etapa a única atividade realizada foi a adição dos valores de predição como colunas, aplicados a todos os dados dos *Data Frames* e não mais em apenas conjuntos de valores distintos, como apresentados na Figura 79 e 80.

Vale ressaltar que os primeiros “*n*” valores, com “*n*” sendo os *time steps* utilizados na criação dos modelos, receberam o valor 0 por conta de não possuírem dados passados suficientes para a predição, como apresentado na Figura 81.

Figura 79 – Plotagem dos modelos aplicados ao período estudado.

```

fig = go.Figure()
fig.add_trace(go.Scatter(x=np.array(range(0,len(y_test_cons))),
                        y=scaler.inverse_transform(y_test_cons).flatten(),
                        mode='lines',
                        name='Consumo de Gás - Real (Test)',
                        opacity=0.8,
                        line=dict(color='black', width=1)
                        ))
fig.add_trace(go.Scatter(x=np.array(range(0,len(pred_test_cons))),
                        y=scaler.inverse_transform(pred_test_cons).flatten(),
                        mode='lines',
                        name='Consumo de Gás - Predito (Test)',
                        opacity=0.8,
                        line=dict(color='red', width=1)
                        ))

# Alterar a cor de fundo do gráfico
fig.update_layout(dict(plot_bgcolor = 'white'))

# Atualizar linhas de eixos
fig.update_xaxes(showgrid=True, gridwidth=1, gridcolor='lightgrey',
                zeroline=True, zerolinewidth=1, zerolinecolor='lightgrey',
                showline=True, linewidth=1, linecolor='black',
                title='Observation'
                )

fig.update_yaxes(showgrid=True, gridwidth=1, gridcolor='lightgrey',
                zeroline=True, zerolinewidth=1, zerolinecolor='lightgrey',
                showline=True, linewidth=1, linecolor='black',
                title='m³'
                )

# Definir título da figura
fig.update_layout(title=dict(text="Consumo de Gás Nitrogênio",
                             font=dict(color='black')),
                 legend=dict(orientation="h", yanchor="bottom", y=1.02, xanchor="right", x=1)
                 )

fig.show()

```

Fontes: Autor (2022)

Figura 80 – Criação da coluna de valores preditos.

```

# Com a configuração atual, alimentamos 1 horas de dados e obtemos a previsão para a hora seguinte
# Queremos criar uma matriz que contenha pedaços de 1 horas compensados por uma hora de cada vez
# Isso é para que possamos fazer uma previsão para todas as horas nos dados, em vez de a cada 1 horas
# CONSUMO
X_every_cons=df_cons[['Consumo de N2 (m³) - Produção Refinaria']]
X_every_cons=scaler.transform(X_every_cons)
for i in range(0, len(X_every_cons)-time_step_cons):
    if i==0:
        X_comb_cons=X_every_cons[i:i+time_step_cons]
    else:
        X_comb_cons=np.append(X_comb_cons, X_every_cons[i:i+time_step_cons])
X_comb_cons=np.reshape(X_comb_cons, (math.floor(len(X_comb_cons)/time_step_cons), time_step_cons, 1))
print(X_comb_cons.shape)
# Use os dados reformulados para fazer previsões e adicioná-los novamente ao dataframe
# np.zeros(time_step) - Defina os primeiros 1 números como 0, pois não temos dados para prever
df_cons['Consumo_predição'] = np.append(np.zeros(time_step_cons), scaler.inverse_transform(model_cons.predict(X_comb_cons)))

# Pressão
X_every_pre=df_pre[['Pressão (bar) - Produção Refinaria']]
X_every_pre=scaler.transform(X_every_pre)
for i in range(0, len(X_every_pre)-time_step_pre):
    if i==0:
        X_comb_pre=X_every_pre[i:i+time_step_pre]
    else:
        X_comb_pre=np.append(X_comb_pre, X_every_pre[i:i+time_step_pre])
X_comb_pre=np.reshape(X_comb_pre, (math.floor(len(X_comb_pre)/time_step_pre), time_step_pre, 1))
print(X_comb_pre.shape)
df_pre['Pressão_predição'] = np.append(np.zeros(time_step_pre), scaler.inverse_transform(model_pre.predict(X_comb_pre)))

# Temperatura
X_every_temp=df_temp[['Temperatura (°C) - Produção Refinaria']]
X_every_temp=scaler.transform(X_every_temp)
for i in range(0, len(X_every_temp)-time_step_temp):
    if i==0:
        X_comb_temp=X_every_temp[i:i+time_step_temp]
    else:
        X_comb_temp=np.append(X_comb_temp, X_every_temp[i:i+time_step_temp])
X_comb_temp=np.reshape(X_comb_temp, (math.floor(len(X_comb_temp)/time_step_temp), time_step_temp, 1))
print(X_comb_temp.shape)
df_temp['Temperatura_predição'] = np.append(np.zeros(time_step_temp), scaler.inverse_transform(model_temp.predict(X_comb_temp)))

```

Fontes: Autor (2022)

Figura 81 – Data frame de Consumo, com valores preditos.

	time	Consumo de N2 (m³) - Produção Refinaria	Consumo_predição
0	2022-08-25 00:00:00	17133.0	0.000000
1	2022-08-25 01:00:00	17135.0	17166.412109
2	2022-08-25 02:00:00	17137.0	17168.421875
3	2022-08-25 03:00:00	17139.0	17170.429688
4	2022-08-25 04:00:00	17142.0	17172.437500
...
1264	2022-10-16 19:00:00	30051.0	30128.353516
1265	2022-10-16 20:00:00	30053.0	30140.404297
1266	2022-10-16 21:00:00	30056.0	30142.412109
1267	2022-10-16 22:00:00	30082.0	30145.425781
1268	2022-10-16 23:00:00	30107.0	30171.539062

1269 rows × 3 columns

Fontes: Autor (2022)

3.4.9 *Previsão dos Parâmetros*

A finalização do estudo foi realizada por meio de uma operação que busca justamente trazer uma maior previsibilidade ao sistema de manuseio de gás nitrogênio e proporcionar aos operadores da área de refinaria a possibilidade de realizar tomadas de decisões da forma mais eficiente e rápida possível, o que muitas vezes pode evitar erros, gastos e demais problemas na companhia.

Sobre isso, temos então que a próxima etapa está focada na realização da previsão dos valores de cada uma das principais variáveis de processo. Especificamente, o intuito dessa etapa está em prever 2 horas a frente do período das amostras, como mostra a Figura 82.

Para esse objetivo, foi desenvolvido uma sequência de funções que seguem como base a metodologia utilizada para realizar as análises preditivas. De forma geral, o processo em questão continua utilizando um aglomerado de “ n ” *time steps* para tentar obter o valor “ $n+1$ ”, onde, à medida que novos valores forem sendo adquiridos, estes por sua vez auxiliaram na obtenção dos próximos parâmetros. No entanto, vale ressaltar, que como este é um procedimento que usa valores não reais, este método possui a limitação de que os erros serão passados de forma acumulativa para cada previsão posterior, logo faz-se necessário prever o mínimo de tempo futuro possível.

Figura 82 – Utilizar a última sequência de dados para prever valores futuros.

```

# Vamos pegar a última sequência nos dados para iniciar as previsões
inputs_cons=X_comb_cons[-1:]
# Criar lista vazia
pred_list_cons = []
# Faça um loop 2 vezes para criar previsões para o próximo dia
for i in range(2):
    pred_list_cons.append(list(model_cons.predict(inputs_cons)[0])) # Gerar previsão e adicioná-la à lista
    inputs_cons = np.append(inputs_cons[:,1:,:],[[pred_list_cons[i]]],axis=1) # Elimine a previsão mais antiga e acrescente a mais recente
# Crie um dataframe contendo 2 horas a partir de 24/10/2022
newdf_cons=pd.DataFrame(pd.date_range(start='2022-10-24 23:00:00', periods=2, freq='H'), columns=['time'])
# Adicionar 2 horas de previsão do modelo da etapa anterior
newdf_cons['Consumo_predição']=scaler.inverse_transform(pred_list_cons)
# Concatenar o dataframe original contendo dados de Canberra e o novo contendo previsões para os próximos 2 horas
df2_cons=pd.concat([df_cons, newdf_cons], ignore_index=False, axis=0, sort=False)

#PARA PRESSÃO
inputs_pre=X_comb_pre[-1:]
pred_list_pre = []
for i in range(2):
    pred_list_pre.append(list(model_pre.predict(inputs_pre)[0]))
    inputs_pre = np.append(inputs_pre[:,1:,:],[[pred_list_pre[i]]],axis=1)
newdf_pre=pd.DataFrame(pd.date_range(start='2022-10-24 23:00:00', periods=2, freq='H'), columns=['time'])
newdf_pre['Pressão_predição']=scaler.inverse_transform(pred_list_pre)
df2_pre=pd.concat([df_pre, newdf_pre], ignore_index=False, axis=0, sort=False)

#PARA TEMPERATURA
inputs_temp=X_comb_temp[-1:]
pred_list_temp = []
for i in range(2):
    pred_list_temp.append(list(model_temp.predict(inputs_temp)[0]))
    inputs_temp = np.append(inputs_temp[:,1:,:],[[pred_list_temp[i]]],axis=1)
newdf_temp=pd.DataFrame(pd.date_range(start='2022-10-24 23:00:00', periods=2, freq='H'), columns=['time'])
newdf_temp['Temperatura_predição']=scaler.inverse_transform(pred_list_temp)
df2_temp=pd.concat([df_temp, newdf_temp], ignore_index=False, axis=0, sort=False)

```

Fontes: Autor (2022)

Os valores calculados serão apresentados de forma gráficas através das funções da Figura 83.

Figura 83 – Plotagem do modelo com duas horas futuras.

```

# Consumo
fig = go.Figure()
fig.add_trace(go.Scatter(x=df2_cons['time'][-1000:],
                        y=df2_cons['Consumo de N2 (m³) - Produção Refinaria'][-1000:],
                        mode='lines',
                        name='Consumo de Gás - Real (Test)',
                        opacity=0.8,
                        line=dict(color='black', width=1)
                        ))
fig.add_trace(go.Scatter(x=df2_cons['time'][-1000:],
                        y=df2_cons['Consumo_predição'][-1000:],
                        mode='lines',
                        name='Consumo de Gás - Predito (Test)',
                        opacity=0.8,
                        line=dict(color='red', width=1)
                        ))

fig.update_layout(dict(plot_bgcolor = 'white'))

fig.update_xaxes(showgrid=True, gridwidth=1, gridcolor='lightgrey',
                zeroline=True, zerolinewidth=1, zerolinecolor='lightgrey',
                showline=True, linewidth=1, linecolor='black',
                title='Observation'
                )

fig.update_yaxes(showgrid=True, gridwidth=1, gridcolor='lightgrey',
                zeroline=True, zerolinewidth=1, zerolinecolor='lightgrey',
                showline=True, linewidth=1, linecolor='black',
                title='m³'
                )

fig.update_layout(title=dict(text="Consumo Total de Gás Nitrogênio",
                             font=dict(color='black')),
                 legend=dict(orientation="h", yanchor="bottom", y=1.02, xanchor="right", x=1)
                 )

fig.show()

```

Fontes: Autor (2022)

4 RESULTADOS E DISCUSSÃO

Neste tópico será discutido e revisado os resultados com a aplicação dos modelos de redes neurais recorrentes nas principais variáveis de processos do setor de refinaria, em uma empresa produtora de margarinas. Faz-se necessário a instrução de ocultar alguns dados por conta de sigilo de mercado.

4.1 Precisão do modelo

Foram obtidos os seguintes resultados, referentes ao R^2 , para o consumo, pressão e temperatura da refinaria, ilustrados na Figura 84. Observa-se que tanto o consumo quanto a temperatura do nitrogênio, possuem valores satisfatórios para o R^2 , diferentemente da pressão, que apresentou um valor negativo, deixando claro que o modelo não conseguiu se adequar corretamente a essa série temporal.

O modelo de consumo, por se tratar de uma série temporal simples e de característica linear, resultou em um R^2 de 99,95%. Acompanhando o bom resultado obtido pelo consumo, o modelo de temperatura nos forneceu o valor de 69,62%. Por fim, em contraponto com os bons modelos apresentados anteriormente, temos que o R^2 apresentado pelo modelo de regressão dos valores de pressão, foi de -5,35%.

Figura 84 – Valores dos R^2 .

```

=====Consumo=====
30/30 [=====] - 0s 725us/step
16/16 [=====] - 0s 801us/step
R²: 99.95222840200458 %
=====Pressão=====
8/8 [=====] - 0s 1ms/step
5/5 [=====] - 0s 1ms/step
R²: -5.349777958565416 %
=====Temperatura=====
4/4 [=====] - 0s 1ms/step
3/3 [=====] - 0s 2ms/step
R²: 69.62495603358039 %

```

Fontes: Autor (2022)

Além dos valores de R^2 , também foram geradas fichas técnicas com algumas das principais informações atreladas às redes neurais, mostradas nas Figuras 85, 86 e 87. Dentre as informações temos: números de *nós* em cada camada, valores de pesos e constantes, e por fim os erros médios quadráticos para os conjuntos de treino e teste.

Figura 85 – Resumo do modelo de previsão de Consumo de Gás Nitrogênio.

```

----- Resumo do modelo - Consumo -----
Model: "First-RNN-Model"

Layer (type)              Output Shape              Param #
-----
Hidden-Recurrent-Layer (SimpleRNN)  (None, 4)                 24
Hidden-Layer (Dense)          (None, 4)                 20
Output-Layer (Dense)         (None, 1)                 5

-----
Total params: 49
Trainable params: 49
Non-trainable params: 0

----- Pesos e bias -----
Observe que o último parâmetro em cada camada é bias, enquanto o restante são pesos

Hidden-Recurrent-Layer
[[ 1.16495765e-04 -7.65892744e-01  9.42511082e-01 -2.62944490e-01]]
[[-0.99242747 -0.05879172  0.10158309  0.03622133]
 [-0.04543728  0.94354486  0.19600312 -0.26313734]
 [ 0.03405096 -0.32340094  0.44333264 -0.8352895 ]
 [-0.10891997  0.04096861 -0.8687461  -0.481392  ]]
 [ 5.7193363e-05  1.0143250e-02 -1.6721684e-01  2.9838940e-02]

Hidden-Layer
[[ 0.18814252 -0.26357982  0.01089358  0.4974151 ]
 [-0.7959585  -0.15433751  0.28120574  0.5303181 ]
 [-0.22045697 -0.73347545  0.24052027 -0.8006188 ]
 [-0.3441668  -0.5208378  0.0431308  0.3096674 ]]
 [ 0.02402887 -0.00448198  0.03606964 -0.01067688]

Output-Layer
[[ 8.1061739e-01]
 [-2.0345695e-01]
 [ 1.5161282e-04]
 [-4.1443548e-01]]
 [0.04357284]

----- Avaliação dos dados de treinamento -----
MSE: 1.1876503236386677e-06

----- Avaliação em dados de teste -----
MSE: 4.523125466445112e-06

```

Fontes: Autor (2022)

Pela imagem da Figura 85, que nos mostra um resumo de algumas variáveis de processos da rede neural, pode-se avaliar, na parte inferior da figura, que estão descritos os valores dos erros quadráticos médios (MSE) aplicados aos conjuntos de dados de treinamento e teste, que possuem respectivamente os valores de $1,19 \times 10^{-6}$ e $4,52 \times 10^{-6}$. Estes valores ajudam a confirmar, juntamente com o valor de R^2 , o quão bem aplicado está o modelo ao conjunto estudado.

Para os valores de pressão do N2 apresentados na Figura 86, temos que os valores dos erros quadráticos para os dados de treinamento e teste foram de aproximadamente 0,05. Importante perceber que mesmo com resultados ruins de R^2 , os valores preditos pelo modelo de pressão, não possuem altos erros médios como seria esperado.

Figura 86 – Resumo do modelo de previsão da Pressão de Gás Nitrogênio.

```

----- Resumo do modelo - Pressão -----
Model: "First-RNN-Model"
-----
Layer (type)          Output Shape          Param #
-----
Hidden-Recurrent-Layer (SimpleRNN) (None, 4)             24
Hidden-Layer (Dense)   (None, 4)             20
Output-Layer (Dense)  (None, 1)             5
-----
Total params: 49
Trainable params: 49
Non-trainable params: 0
-----

----- Pesos e bias -----
Observe que o último parâmetro em cada camada é bias, enquanto o restante são pesos

Hidden-Recurrent-Layer
[[[-0.84540355  1.1855782  0.4939805  1.0911021 ]]
 [[-0.7246788  -0.49876904 -0.24235038  0.20579985]
 [ 0.08013432 -0.50560194  0.92051905  0.34619835]
 [-0.08910147 -0.46501052  0.10064333 -0.9810641 ]
 [-0.667161    0.6854776  0.6174917  -0.08649961]]
 [-0.0476834  -0.08533301 -0.07820649 -0.1138112 ]]

Hidden-Layer
[[[ 0.5571011  0.69420904 -0.30659556 -0.90474075]
 [ 0.6398247 -0.22449313 -0.06413058  0.7938525 ]
 [-0.69199026 -0.6102625  0.48034316 -0.25758466]
 [-0.74571675 -0.9480499  -0.07200513 -0.35021856]]
 [0.0877744  0.07556356  0.14471261  0.09479759]]

Output-Layer
[[[-0.7402463 ]
 [ 0.96987396]
 [ 0.33479363]
 [ 1.0569634 ]]
 [0.07730773]]

----- Avaliação dos dados de treinamento -----
MSE: 0.0506723512107129

----- Avaliação em dados de teste -----
MSE: 0.055130021201618876

```

Fontes: Autor (2022)

Por fim, os valores referentes aos MSE's dos datasets de treino e teste, para os dados de temperatura de utilização do nitrogênio no setor de refinaria, são 0,0077 (Treinamento) e 0,01 (Teste). Os valores podem ser visualizados na Figura 87 a seguir.

Figura 87 – Resumo do modelo de previsão da Temperatura de Gás Nitrogênio.

```

----- Resumo do modelo - Temperatura -----
Model: "First-RNN-Model"
-----
Layer (type)              Output Shape              Param #
-----
Hidden-Recurrent-Layer (SimpleRNN)  (None, 4)                 24
Hidden-Layer (Dense)          (None, 4)                 20
Output-Layer (Dense)         (None, 1)                 5
-----
Total params: 49
Trainable params: 49
Non-trainable params: 0
-----

----- Pesos e bias -----
Observe que o último parâmetro em cada camada é bias, enquanto o restante são pesos

Hidden-Recurrent-Layer
[[[-1.0659695 -0.37312296  1.3750917  0.8230285 ]
 [ 0.7077616  0.09279998  0.8086453  0.50992733]
 [-0.10208269  0.06539377 -0.21865101  1.1529297 ]
 [ 0.32548818 -0.96153545 -0.23864879 -0.30194578]
 [ 0.3194864  0.63921684  0.13966143 -0.11695559]]
 [ 0.08362214 -0.04321015 -0.03175514 -0.0994722 ]
Hidden-Layer
[[[ 0.738641  0.319651  0.21780524  0.8063702 ]
 [ 0.32187673 -0.20129606  0.61870354  0.47653258]
 [-0.203802  -0.9754483  -0.42545244 -0.43151137]
 [-0.90424955 -0.4436342  0.5304242  0.7920536 ]]
 [-0.00050775 -0.04215204 -0.18672946 -0.05486502]
Output-Layer
[[[-0.3132581 ]
 [-0.65164596]
 [-0.35620382]
 [-0.07192151]]
 [0.15700006]

----- Avaliação dos dados de treinamento -----
MSE: 0.007775541095468507

----- Avaliação em dados de teste -----
MSE: 0.011936690459048388

```

Fontes: Autor (2022)

4.2 Valores Preditos

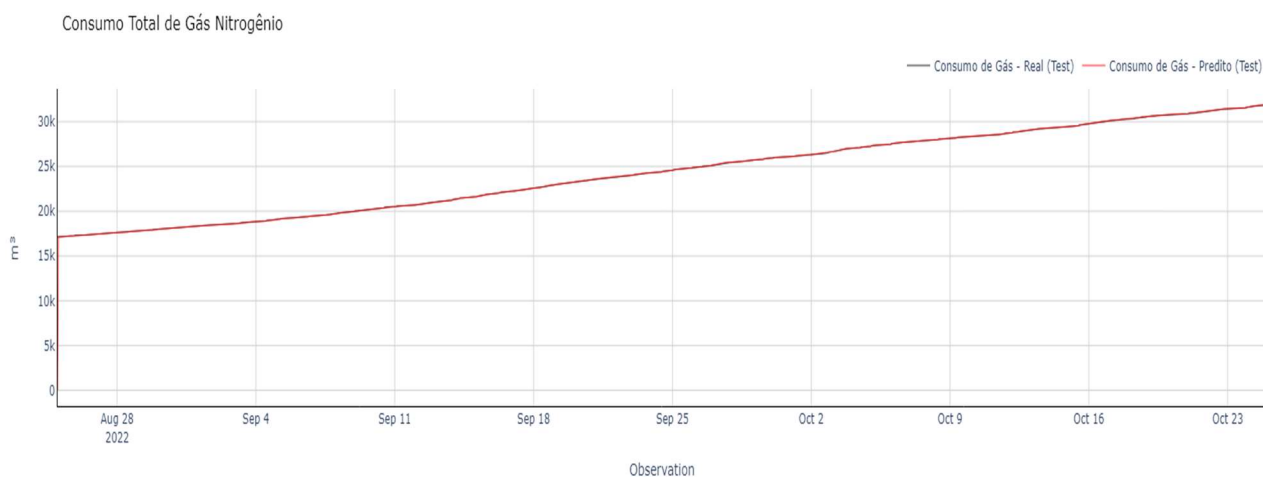
Como já descrito anteriormente, para ter uma melhor noção do comportamento da predição perante os dados coletados inicialmente pelos sensores, foram plotados ambos os conjuntos em uma mesma sequência de gráficos, apresentados nas Figuras 88, 89 e 90. Vale

relembrar que, como falado na Seção 3.4, as amostras iniciais de predição, apresentam valores nulos.

Outro ponto interessante de se perceber vem do fato de que as predições para a pressão de N₂, não está se apresentando de forma completamente displicente dos números reais, ao contrário do que se imaginava. O que corrobora com duas possibilidades principais que possam ser o motivo de que sua baixa precisão:

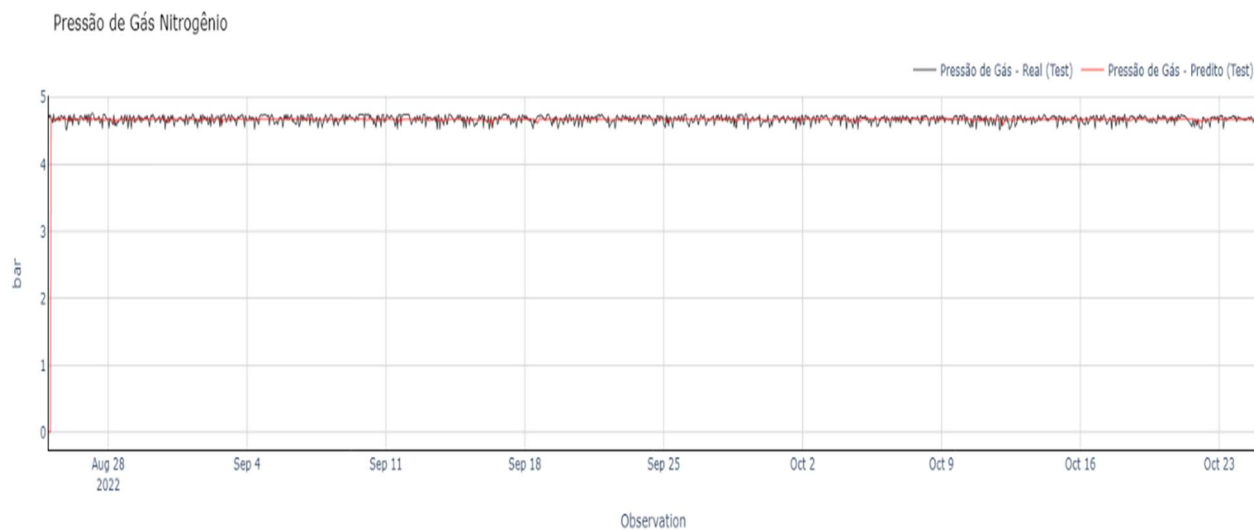
- O modelo RNN utilizado não ser o mais adequado para seu estudo. Logo, precisaria trocar para outro método.
- O número de iterações não foi o suficiente para que o modelo pudesse aprender com a série temporal. Necessitando de maior poder de processamento.

Figura 88 – Desempenho do modelo de consumo, com os dados dos sensores.



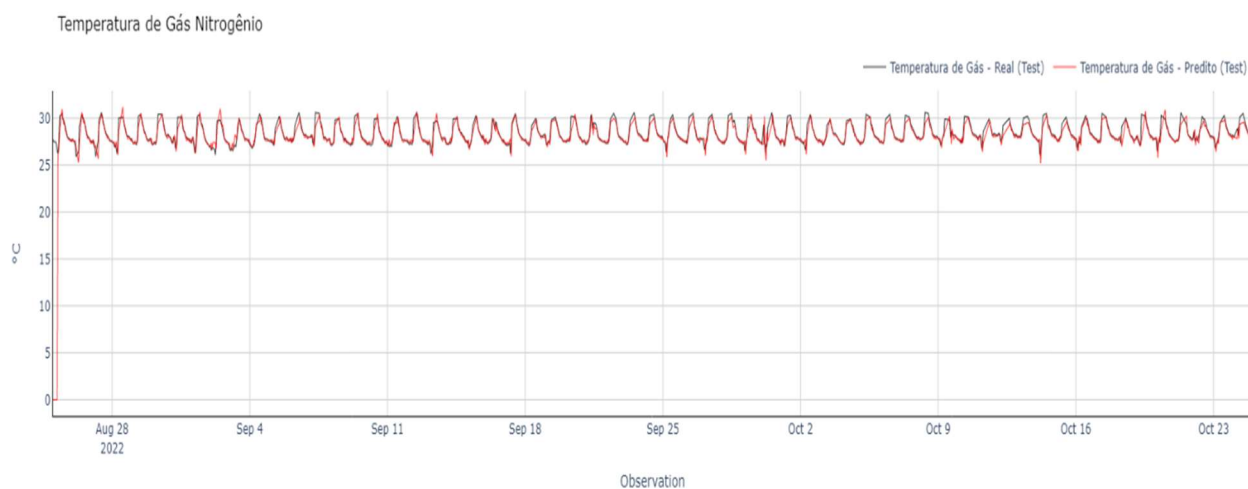
Fontes: Autor (2022)

Figura 89 – Desempenho do modelo de pressão, com os dados dos sensores.



Fontes: Autor (2022)

Figura 90 – Desempenho do modelo de temperatura, com os dados dos sensores.



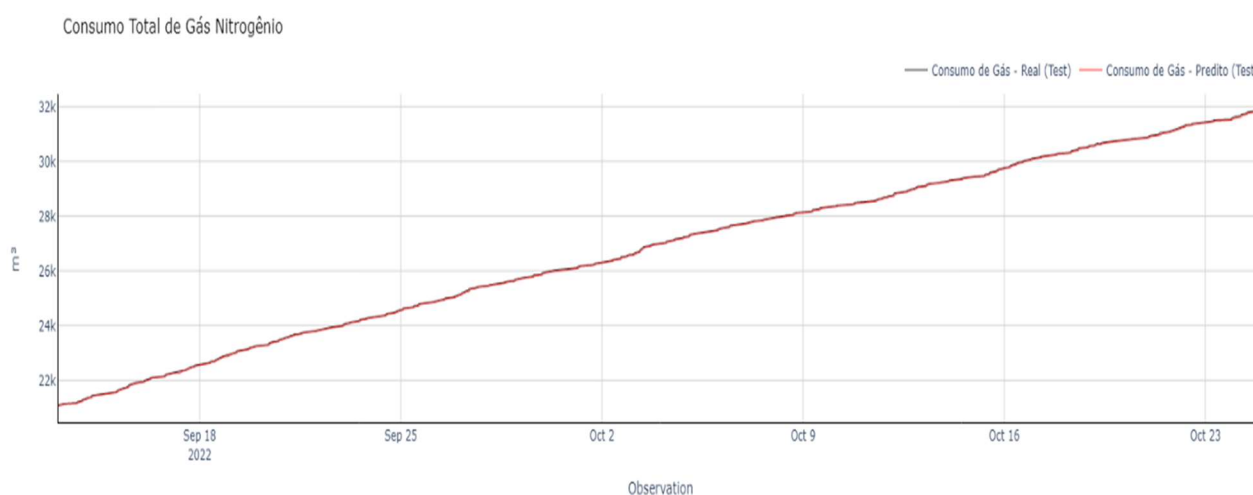
Fontes: Autor (2022)

4.3 Previsões das variáveis

Para finalizar o trabalho, temos as representações gráficas dos dados de previsão dos parâmetros de gás nitrogênio, aplicados para a previsão de 2 horas futuras. Em cada Figura existe um gráfico portando os valores reais e previstos para todo o período amostral, e outro gráfico com os mesmos valores, porém com uma visão focada nos dados futuros.

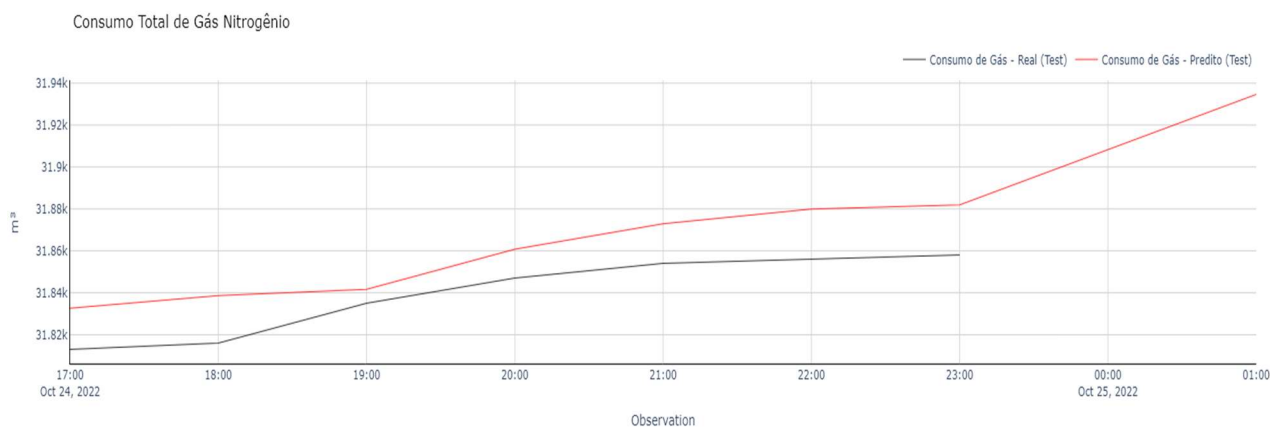
Nas Figuras 91 e 92, podemos avaliar a curva gerada pelo modelo de rede neural recorrente estudado, aplicado aos valores coletados de consumo de gás nitrogênio, que está representado pela linha vermelha dos gráficos. Também nos gráficos anteriores temos representado pela linha preta, os valores reais do metro cúbico coletado pelos equipamentos da companhia.

Figura 91 – Previsão de 2 horas futuras, para o Consumo em escala completa.



Fontes: Autor (2022)

Figura 92 – Previsão de 2 horas futuras, para o Consumo em escala reduzida.



Fontes: Autor (2022)

O primeiro gráfico apresentado nos apresenta uma versão expandida dos dois conjuntos de dados citados, para que possamos realizar uma avaliação visual da acurácia dos valores

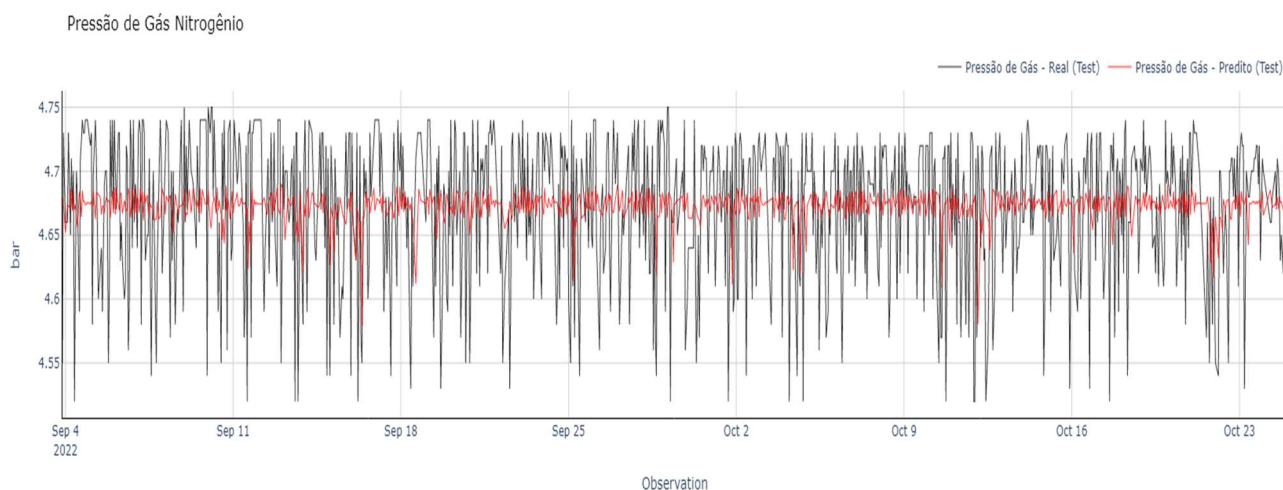
previstos. Já na segunda imagem, temos uma versão ampliada do mesmo conjunto de dados, onde está focada em apresentar os valores de consumo para duas horas futuras.

Temos que, os valores reais coletados vão até o dia 24 de Outubro de 2022 às 23 horas, já as amostras previstas nos fornecem o valor de 31,901 mil metros cúbicos à meia noite do dia 25 de Outubro, bem como o valor de 31,934 mil metros cúbicos, para as 01 horas do mesmo dia.

Com relação aos resultados do modelo de consumo, pode-se averiguar que o comportamento da regressão está quase que perfeitamente ajustada aos valores reais e podendo ser aplicada quando necessário.

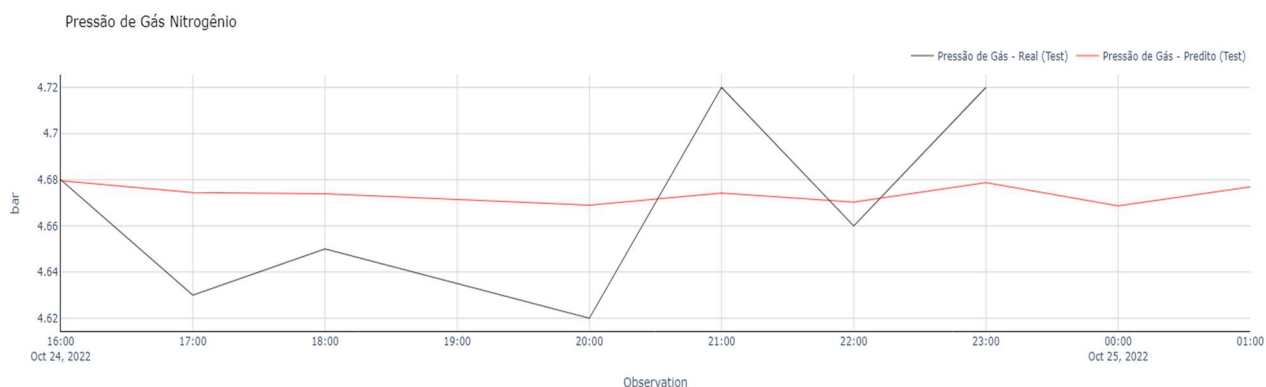
De acordo com os gráficos de pressão das Figuras 93 e 94, pode-se avaliar que o modelo buscou seguir o comportamento da linha de dados reais, mais até do que o esperado, tendo em vista que seu valor de R^2 foi muito baixo, no entanto é perceptível que a variância do processo ainda é muito grande para poder validar o modelo de forma positiva.

Figura 93 – Previsão de 2 horas futuras, para a Pressão em escala completa.



Fontes: Autor (2022)

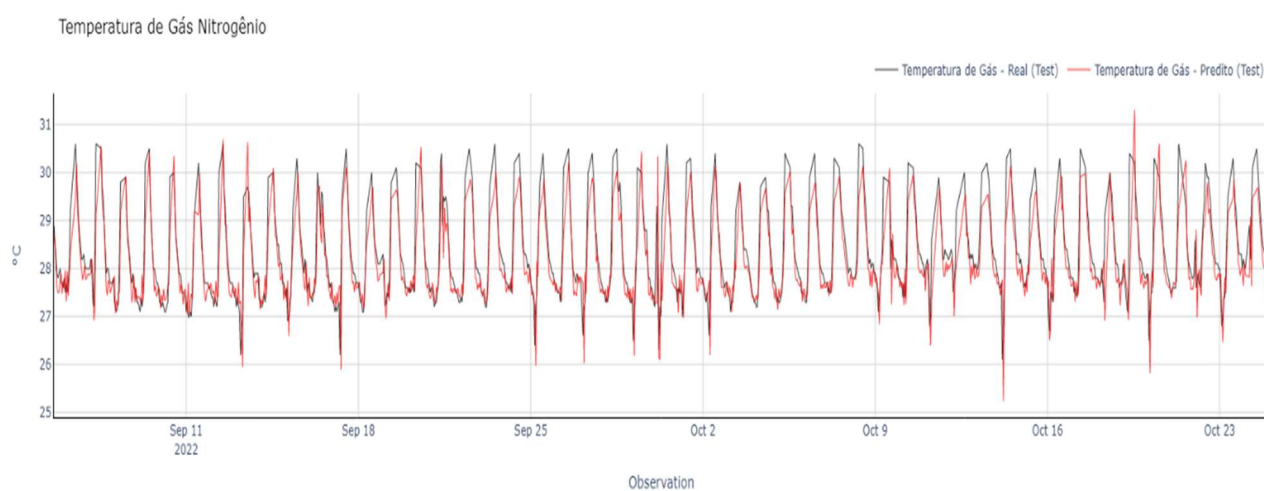
Figura 94 – Previsão de 2 horas futuras, para a Pressão em escala reduzida.



Fontes: Autor (2022)

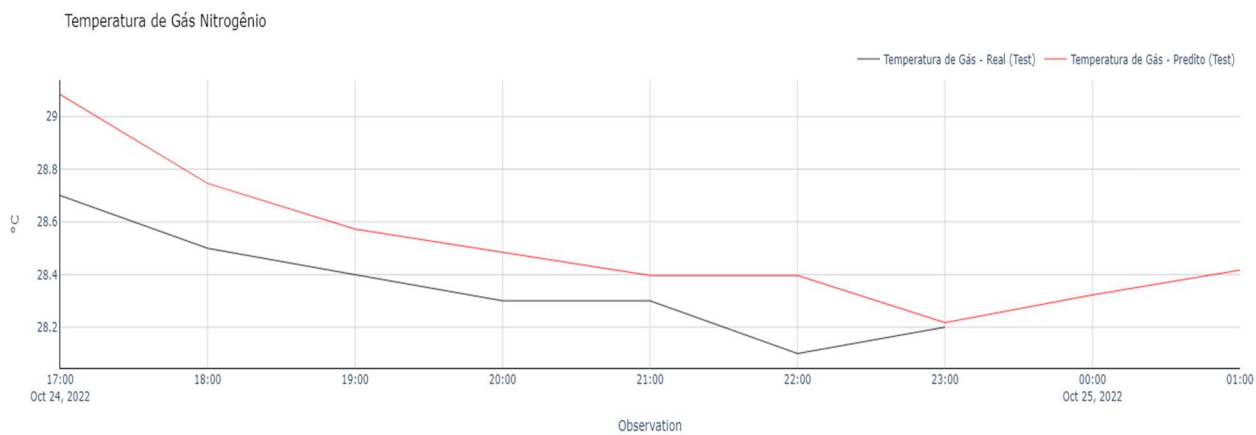
Para os valores de temperaturas das Figuras 95 e 96, as amostras previstas nos fornecem o valor de $28,32\text{ }^{\circ}\text{C}$ à meia noite do dia 25, e $28,41\text{ }^{\circ}\text{C}$ para às 01 horas. Pode-se inferir com estes resultados que apesar de possuir uma margem de erro maior, se comparado ao modelo de consumo e devido a uma maior complexidade das medidas de temperatura, não é errado concluir que a regressão se adequou bem às amostras estudadas e que poderá ser aplicada em quaisquer projetos, como medida de controle na variação dos processos.

Figura 95 – Previsão de 2 horas futuras, para a Temperatura em escala completa.



Fontes: Autor (2022)

Figura 96 – Previsão de 2 horas futuras, para a Temperatura em escala reduzida.



Fontes: Autor (2022)

5 CONSIDERAÇÕES FINAIS

Neste trabalho foi desenvolvida uma implementação de uma ferramenta para a previsão dos valores condicionais da utilização de gás nitrogênio para uma empresa produtora de margarinas. Ao longo deste desenvolvimento, um tratamento da base de dados foi necessário, visto que a base disponibilizada pela companhia apresentava uma grande quantidade de outliers que viriam a afetar o desempenho da rede neural.

Com relação ao processo de obtenção dos valores, os mesmos foram obtidos utilizando sensores da empresa que captam essas informações e armazenam os dados de forma digital, porém não fornecendo quaisquer estudo ou análise nos mesmos.

Para trabalhos futuros, seria interessante um levantamento sobre outras técnicas para desenvolvimento de modelos de redes neurais, que se adequassem melhor ao comportamento das variáveis estudadas.

O modelo de rede neural usado, foi o RNN, devido à sua característica de recorrência que o faz um modelo ideal para aplicação em séries temporais, todavia alguns modelos como LSTM e GRU, também partilham de características parecidas e poderiam ser utilizados em teste futuros.

Os modelos criados obtiveram valores de R^2 variáveis. Temos que pra o consumo de gás, o modelo possuiu um R^2 de 99,95%, que se classifica como um ótimo valor e nos diz que o modelo se adequou quase que completamente a série temporal. O modelo de temperatura, resultou em um R^2 de 69,62%, que também pode ser classificado como um bom modelo, e significando que a rede neural irá funcionar de forma esperada, com poucos erros. Já o modelo de previsão dos valores de pressão, forneceu um R^2 de -5,35%, o que de acordo com a literatura, pode-se concluir que o modelo aplicado não pôde aprender como o comportamento da variável funciona.

Desta forma, o presente trabalho obteve sucesso em obter os valores de duas horas futuras, para as variáveis de consumo e temperatura de gás nitrogênio, podendo o mesmo ser aplicado tanto em análises administrativas na empresa, com intuito de auxiliar na tomada de decisão dos gestores da área, pois estes dados permitirão um vislumbre do desempenho do setor, evitando falhas nos processos e gastos desnecessários.

No entanto, o projeto falhou em realizar a previsão correta dos dados de pressão do gás, portanto o mesmo não poderá ser usado nas atividades de manutenção do desempenho do sistema, como falado anteriormente. Para que o modelo possa vir a funcionar de forma devida, faz-se necessário o uso de uma maior quantidade de dados da companhia e um maior poder de processamento de informações, que permitirão a rede neural, se adequar ao comportamento dos dados.

REFERÊNCIAS

NITROGÊNIO. Toda matéria, 2022. Disponível em:

<<https://www.todamateria.com.br/nitrogenio/>>. Acesso em: 01 de Novembro de 2022.

AZANK, Felipe. Como avaliar seu modelo de regressão. Medium, 2020. Disponível em:

<<https://medium.com/turing-talks/como-avaliar-seu-modelo-de-regressão-c2c8d73dab96>>.

Acesso em: 01 de Novembro de 2022.

USO E APLICAÇÕES DO NITROGÊNIO. Omega air, 2022. Disponível em:

<<https://www.omega-air.pt/noticias/noticias/usos-e-aplicacoes-do-nitrogenio>>. Acesso em: 01 de Novembro de 2022.

INDÚSTRIA 4.0: ENTENDA SEUS CONCEITOS E FUNDAMENTOS. Portal da Indústria, 2022. Disponível em: <<https://www.portaldaindustria.com.br/industria-de-a-z/industria-4-0/>>.

Acesso em: 01 de Novembro de 2022.

O QUE É INDÚSTRIA 4.0?. FluiPress, 2022. Disponível em: <<https://fluiPress.com.br/industria-4-0/o-que-e-industria-4-0/>>. Acesso em: 01 de Novembro de 2022.

INDÚSTRIA 4.0 – O QUE É E COMO ELA DEVE AFETAR NOSSAS VIDAS. Wert

ambiental, 2019. Disponível em: <https://wertambiental.com.br/2019/01/15/industria_4-0/>.

Acesso em: 01 de Novembro de 2022.

CARDOSO, Marcelo D.C. INDÚSTRIA 4.0: a quarta revolução industrial. Universidade Tecnológica Federal do Paraná. CURITIBA – PA, 2016.

FREITAS, Amanda D.P.P. ANÁLISE BIBLIOMÉTRICA DA PRODUÇÃO CIENTÍFICA SOBRE INDÚSTRIA 4.0. Universidade Federal de Uberlândia. UBERLÂNDIA – MG, 2018.

JUNIOR, Geraldo T; SALTORATO, Patrícia. IMPACTOS DA INDÚSTRIA 4.0 NA

ORGANIZAÇÃO DO TRABALHO: UMA REVISÃO SISTEMÁTICA DA LITERATURA.

Revista Produção Online, Vol. 18, N. 2, P. 743 – 769, FLORIANÓPOLIS – SC, Junho de 2018.

UTILIZAÇÃO DE BIG DATA NA INDÚSTRIA 4.0. Automação Industrial.

, 2017. Disponível em: <<https://www.automacaoindustrial.info/utilizacao-de-big-data-na-industria-4-0/>>. Acesso em: 01 de Novembro de 2022.

KOIKE, Tiago. O Big Data na Indústria 4.0: Qual a sua importância?. Pollux, 2018. Disponível

em: <<https://pollux.com.br/blog/big-data-na-industria-4-0-qual-sua-importancia/>>. Acesso em: 01 de Novembro de 2022.

HOME. ST-One, 2022. Disponível em: <<https://st-one.io/>>. Acesso em: 01 de Novembro de 2022.

BIG DATA. ATRA, 2022. Disponível em: <<https://www.atrainformatica.com.br/big-data/>>. Acesso em: 01 de Novembro de 2022.

DIGITALIZAÇÃO DE PROCESSOS PRODUTIVOS NA INDÚSTRIA: COMO IMPLEMENTAR?. Petronas Inovação Industrial, 2022. Disponível em: <<https://inovacaoindustrial.com.br/digitalizacao-de-processos-produtivos-na-industria-como-implementar/>>. Acesso em: 01 de Novembro de 2022.

INTELIGÊNCIA ARTIFICIAL: O GUIA COMPLETO SOBRE O ASSUNTO. TOTVS, 2022. Disponível em: <<https://www.totvs.com/blog/inovacoes/o-que-e-inteligencia-artificial/>>. Acesso em: 01 de Novembro de 2022.

CONCEITOS BÁSICOS SOBRE MACHINNE LEARNING. 4.Linux, 2022. Disponível em: <<https://blog.4linux.com.br/conceitos-basicos-sobre-machine-learning/>>. Acesso em: 01 de Novembro de 2022.

KENJI, Bruno. Machine Learning para Leigos. Venturus, 2019. Disponível em: <<https://www.venturus.org.br/machine-learning-para-leigos/>>. Acesso em: 01 de Novembro de 2022.

INTELIGÊNCIA ARTIFICIAL E DEEP LEARNING. UFOP, 2019. Disponível em: <<http://www2.decom.ufop.br/imobilis/inteligencia-artificial-e-deep-learning/>>. Acesso em: 01 de Novembro de 2022.

O QUE É UMA REDE NEURAL?. AWS, 2022. Disponível em: <<https://aws.amazon.com/pt/what-is/neural-network/>>. Acesso em: 01 de Novembro de 2022.

REDES NEURASIS ARTIFICIAIS. USP, 2022. Disponível em: <<https://sites.icmc.usp.br/andre/research/neural/>>. Acesso em: 01 de Novembro de 2022.

FÁVERO, Patrícia B; ZOUCAS, Fabio A.M. REDES NEURASIS PARA PREVISÃO DA PRODUÇÃO INDUSTRIAL DE DIFERENTES SEGMENTOS. Produto & Produção, Vol. 17, N. 2, P. 53 – 70, SANTO ANDRÉ – SP, Junho de 2016.

TURKIEWICZ, Matheus S; FRACAROLLI, Rodrigo L. REDES NEURASIS ARTIFICIAIS: IMPORTÂNCIA DA APLICAÇÃO NA INDÚSTRIA BRASILEIRA. ENEGEP, SANTOS – SP, Outubro de 2019.

SOUZA, Rodrigo C.T.D. Previsão de séries temporais utilizando rede neural treinada por filtro de kalman e evolução diferencial. Universidade Católica do Paraná. CURITIBA – PA, 2008.

MONTEIRO, Lee V; GEMARI, Marcia. Aplicações de Modelos ARIMA e Redes Neurais para Previsão de Demanda em Curto Prazo. Pontifícia Universidade Católica do Paraná.

INTRODUÇÃO A AUTOENCODERS. Deep Learning Book, 2022. Disponível em: <<https://www.deeplearningbook.com.br/introducao-aos-autoencoders/>>. Acesso em: 01 de Novembro de 2022.

CECCON, Denny. Os tipos de redes neurais. IA Expert Academy, 2020. Disponível em: <<https://iaexpert.academy/2020/06/08/os-tipos-de-redes-neurais/>>. Acesso em: 01 de Novembro de 2022.

UM MERGULHO PROFUNDO NAS REDES NEURAIIS RECORRENTES. imaster, 2017. Disponível em: <<https://imasters.com.br/data/um-mergulho-profundo-nas-redes-neurais-recorrentes/>>. Acesso em: 01 de Novembro de 2022.

REDES NEURAIIS RECORRENTES. Deep Learning Book, 2022. Disponível em: <<https://www.deeplearningbook.com.br/redes-neurais-recorrentes/>>. Acesso em: 01 de Novembro de 2022.

O QUE SÃO REDES NEURAIIS RECORRENTES. Didática Tech, 2022. Disponível em: <<https://didatica.tech/como-funcionam-redes-neurais-recorrentes-rnn/>>. Acesso em: 01 de Novembro de 2022.

SANTANA, Luciana M.Q.D. APLICAÇÃO DE REDES NEURAIIS RECORRENTES NO RECONHECIMENTO AUTOMÁTICO DA FALA EM AMBIENTES COM RUÍDOS. Universidade Federal de Sergipe, SÃO CRISTÓVÃO – SE, 2017.

WHAT IS THE DIFFERENCE BETWEEN LSTM AND RNN. Stack Exchange, 2022. Disponível em: <<https://ai.stackexchange.com/questions/18198/what-is-the-difference-between-lstm-and-rnn>>. Acesso em: 01 de Novembro de 2022.

CARACTERÍSTICAS DAS SÉRIES TEMPORAIS. Statplace, 2020. Disponível em: <<https://statplace.com.br/blog/caracteristicas-das-series-temporais/>>. Acesso em: 01 de Novembro de 2022.

PINHEIRO, Nina M. Introdução a Series Temporais – Parte 1. Medium, 2021. Disponível em: <<https://medium.com/data-hackers/series-temporais-parte-1-a0e75a512e72>>. Acesso em: 01 de Novembro de 2022.

SPADINI, Allan S. SÉRIES TEMPORAIS E SUAS APLICAÇÕES. Alura, 2021. Disponível em: <<https://www.alura.com.br/artigos/series-temporais-e-suas-aplicacoes>>. Acesso em: 01 de Novembro de 2022.

MIGON, Hélio. Análise de Séries Temporais. 2022. 59 Slides. Disponível em: <<http://www.dme.ufjf.br/dani/pdf/slidespartefrequentista.pdf>>. Acesso em: 01 de Novembro de 2022.

KUROHANE, Caio. MODELOS ESTATÍSTICOS NA PREVISÃO DE DEMANDA. Imam, 2022. Disponível em: <<https://blog.imam.com.br/modelos-estatisticos-na-previsao-de-demanda/>>. Acesso em: 01 de Novembro de 2022.

HENRIQUES, Carla. Análise de Regressão Linear Simples e Múltipla. 2011. 44 Slides. Disponível em: <[https://edisciplinas.usp.br/pluginfile.php/176802/mod_resource/content/1/Slides - Regressão Linear Simples e Múltipla.pdf](https://edisciplinas.usp.br/pluginfile.php/176802/mod_resource/content/1/Slides-Regressão%20Linear%20Simples%20e%20Múltipla.pdf)>. Acesso em: 01 de Novembro de 2022.

GUIA COMPLETO DE SQL. Devmedia, 2022. Disponível em: <<https://www.devmedia.com.br/guia/guia-completo-de-sql/38314>>. Acesso em: 01 de Novembro de 2022.

SQL: O QUE É E COMO USAR OS PRINCIPAIS COMANDOS BÁSICOS SQL. Betrybe, 2022. Disponível em: <<https://blog.betrybe.com/sql/>>. Acesso em: 01 de Novembro de 2022.

SOUZA, Alex. O que é uma rede neural artificial (código python). Wordpress, 2019. Disponível em: <<https://blogdozouza.wordpress.com/2019/12/09/como-codificar-uma-rede-neural-com-backpropagation-em-python-do-zero/>>. Acesso em: 01 de Novembro de 2022.

PYTHON: O QUE É, COMO USAR, GUIA PRA APRENDER A LINGUAGEM. Betrybe, 2022. Disponível em: <<https://blog.betrybe.com/python/>>. Acesso em: 01 de Novembro de 2022.

