



UNIVERSIDADE FEDERAL DE ALAGOAS
UNIDADE ACADÊMICA CENTRO DE TECNOLOGIA
CURSO DE ENGENHARIA QUÍMICA

GAMALIEL DA SILVA TAVARES

**INFERÊNCIA ESTATÍSTICA NO REATOR DE VAN DE VUSSE POR MEIO DE
*MACHINE LEARNING.***

Maceió

2022

GAMALIEL DA SILVA TAVARES

**INFERÊNCIA ESTATÍSTICA NO REATOR DE VAN DE VUSSE POR MEIO DE
*MACHINE LEARNING.***

Trabalho de Conclusão de Curso apresentado ao curso de Engenharia Química da Universidade Federal de Alagoas como requisito parcial para obtenção do título de Bacharel em Engenharia Química.

Orientador: Prof. Dr. Soraya Lira Alencar.

Coorientador: Prof. Dr. Frede de Oliveira Carvalho.

Maceió

2022

Catálogo na fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico
Bibliotecária: Taciana Sousa dos Santos – CRB-4 – 2062

T231i Tavares, Gamaliel da Silva.
Inferência estatística no reator de Van de Vusse por meio de *machine learning* / Gamaliel da Silva Tavares. – 2022.
39 f. : il. color.

Orientadora: Soraya Lira Alencar.
Coorientador: Frede de Oliveira Carvalho.
Monografia (Trabalho de Conclusão de Curso em Engenharia Química)
– Universidade Federal de Alagoas. Centro de Tecnologia. Maceió, 2022.

Bibliografia: f. 37-39.

1. *Machine Learning*. 2. Sensores virtuais. 3. Reator – Van de Vusse. 4. Rede neural artificial. 5. Indústria 4.0. I. Título.

CDU: 66.0: 004.8

Dedico este trabalho aos meus pais,
que me apoiam e incentivam a continuar firme nessa caminhada.

AGRADECIMENTOS

Em primeiro lugar, agradeço aos meus pais, que me apoiaram desde o começo dessa trajetória e me deram forças com palavras de apoio nos momentos difíceis.

Agradeço ao professor Frede de Oliveira Carvalho e a professora Soraya Lira Alencar, pelo grande incentivo e valiosas contribuições com seus conhecimentos na área de inteligência artificial e cinética de reatores, que nortearam de forma assertiva este trabalho.

Aos amigos, Larissa Rodrigues, Emanuel Alysson, João Paulo e Ana Lara agradeço o companheirismo e a amizade, que foram essenciais nessa trajetória, com vários momentos indescritíveis e essenciais para finalização do curso.

Resumo

Atualmente técnicas relacionadas a inteligência artificial tais como as *Machine Learning* (maquinas de aprendizagem), são aplicadas em diversas áreas, porem na engenharia química sua maior aplicação é no campo de sensores virtuais (*Soft Sensor*) e detectores automáticos de falha. Se trata de algoritmos que possuem estrutura básica extremamente maleável, grande simplicidade e adaptabilidade aos dados trabalhados. O funcionamento de um *Soft Sensor* de maneira geral é uma estimativa de sinal, interpolação e previsão (inferência) que tem o objetivo de gerar uma medição virtual em função de variáveis de fácil medição a fim de substituir uma variável monitorada experimentalmente por análise. O presente trabalho abordou um problema típico na indústria química, que se trata da inferência estatística, onde foi utilizado o reator de Van de Vusse que é um processo bem conhecido na literatura por ter muitas não linearidades e consequente complexidade que dificulta o monitoramento e controle, que será modelado com o auxílio de Redes Neurais Artificiais (*RNA's*), que são modelos computacionais com base no funcionamento do cérebro humano, os quais, por meio da conexão de neurônios artificiais, são capazes de aprender relações complexas entre dados. Os dados para a avaliação da eficiência da *Machine Learning* como sensor virtual ou *Soft Sensor*, foram gerados através da modelagem do reator de Van de Vusse representado matematicamente a partir de seus balanços de massa e energia com os sistemas de equações diferenciais ordinárias que foram resolvidos numericamente pelo método de Euler. Foi observado que a definição do tempo de amostragem é uma etapa importante no processo, visto que beneficia as métricas de desempenho da *Soft Sensor*. Para a abordagem estabelecida a Rede Neural Artificial desempenhou um resultado satisfatório na construção do sensor virtual, se mostrando viável para aplicação em plantas industriais. Tanto a obtenção quanto a utilização dos dados nas RNAs contaram com o auxílio do software *Python*, que disponibiliza gratuitamente bibliotecas para implementação dos algoritmos e vem apresentando um grande interesse da indústria para aplicação da engenharia 4.0.

Palavras-Chave: Soft Sensor; Machine Learning; Indústria 4.0; Reator de Van de Vusse.

Abstract

Currently, technical applications related to artificial intelligence, such as Machine Learning, are applied in several areas, but in chemical engineering its biggest of Soft Sensor and automatic failure detectors. These are algorithms that have an extremely malleable basic structure, great simplicity and adaptability to the data being worked on. The operation of a Soft Sensor in general is a signal estimation, interpolation and prediction (inference) that has the objective of generating a virtual measurement as a function of easy-to-measure variables in order to replace a variable experimentally monitored by analysis. The present work addressed a typical problem in the chemical industry, which is statistical inference, where the Van de Vusse reactor was used, which is a well-known process in the literature for having many nonlinearities and consequent complexity that makes monitoring and control difficult, which will be modeled with the help of Artificial Neural Networks (ANN's), which are computational models based on the functioning of the human brain, which, through the connection of artificial neurons, are able to learn complex relationships between data. Data for the evaluation of the efficiency of Machine Learning as a virtual sensor or Soft Sensor, were generated by modeling the Van de Vusse reactor mathematically represented from its mass and energy balances with systems of ordinary differential equations that were numerically solved by Euler's method. It was observed that the definition of the sampling time is an important step in the process, as it benefits the Soft Sensor performance metrics. For the established approach, the Artificial Neural Network performed a satisfactory result in the construction of the virtual sensor, proving to be viable for application in industrial plants. Both obtaining and using the data in the ANNs relied on the help of Python software, which provides free libraries for implementing the algorithms and has been showing great interest from the industry for the application of engineering 4.0.

Key Word: Soft Sensor; Machine Learning; Industry 4.0; Van de Vusse reactor.

Sumário	
1. INTRODUÇÃO	9
2. OBJETIVOS	11
2.1 Geral	11
2.2 Específicos	11
3. REVISÃO BIBLIOGRÁFICA	12
3.1 Reação de Van de Vusse	12
3.2 Inferência estatística e <i>Soft Sensor</i>	15
3.3 Aprendizado de máquina <i>machine learning</i>	16
3.4 Redes neurais artificiais	17
3.5 Redes neurais multicamadas	19
3.6 Treinamento	19
3.7 Programação em python	21
3.8 Computação em nuvem	21
4. METODOLOGIA	22
4.1 Seleção dos dados de estudo	24
4.2 Tratamento de dados	24
4.2.1 Normalização dos dados	25
4.3 Sinais de entrada	26
4.4 Detalhes do modelo utilizado	26
4.5 Critérios de avaliação	29
4.6 Ambiente de treinamento	29
5. RESULTADOS E DISCUSSÕES	30
5.1 Modelagem do reator de Van de Vusse	30
5.2 Definição do tempo de amostragem	31
5.3 Geração do sinal de entrada	34
5.4 Validação do modelo	35
6. CONCLUSÃO	37
7. REFERÊNCIAS	38

1. INTRODUÇÃO

Atualmente, a capacidade de gerar, armazenar e acessar dados aumentou exponencialmente, com crescendo, também, a demanda por mecanismos de análise e acompanhamento automatizado de processo que visam aumentar a produção, segurança e atenção as questões ambientais. Atualmente análises com técnicas de *Machine Learning* são aplicadas em diversas áreas. Na engenharia química sua maior aplicação é no *Soft Sensor* e detectores automáticos de falha, onde os algoritmos possuem estrutura básica extremamente maleável, grande simplicidade e adaptabilidade aos dados trabalhados.

Na indústria química, *Machine Learning* já possui um espaço estabelecido desde os anos 1990, com a aplicação, por exemplo, de redes neurais para fazer aproximação de 2 processos para fins de controle automático, quimiometria e diagnóstico de falhas. (BAUGHMAN & LIU, 1995; DE SOUZA JR, 1993)

A inferência de variáveis (*Soft Sensor*), detecção e classificação de falhas, previsão de desempenho, aproximação de funções não lineares e identificação de padrões são problemas típicos da indústria e da investigação de fenômenos em geral. Em virtude da dificuldade operacional ou até de um alto custo dos sensores para monitoramento de uma variável desejada, os *Soft Sensor* surgem como uma boa alternativa em relação aos sensores tradicionais.

Nos últimos anos o estudo a respeito de *Soft Sensor* vem crescendo amplamente junto com sua utilização no controle de processos industriais para melhorar a qualidade do produto e garantir a segurança na produção. O funcionamento de um *Soft Sensor* de maneira geral é uma estimativa de sinal, interpolação e previsão (inferência) que tem o objetivo de gerar uma medição virtual em função de variáveis de fácil medição a fim de substituir uma variável monitorada experimentalmente por análise ou de difícil medição.

Segundo Soares (2017), o maior desafio atual com *Machine Learning* na indústria química é a qualidade dos dados. Todos os métodos de *Machine Learning* dependem de muitos dados para um aprendizado satisfatório. Ruídos, falhas nos sensores e temporalidade dos dados são fatores a serem considerados antes de se aplicar *Machine Learning* a engenharia de processos. Vale salientar, que no desenvolvimento de *Soft Sensor*, através de diversas técnicas, incluindo *Machine Learning*, são utilizados dados obtidos a partir da modelagem e simulação do processo, uma vez que dados experimentais são muitas vezes difíceis de se obter ou disponibilizados.

O reator de Van de Vusse é um processo bastante abordado na literatura por ter muitas não linearidades e conseqüente complexidade por apresentar reações em serie e paralelo que dificultam o monitoramento e controle. Reportado, muitas vezes na literatura, como um estudo de caso na construção de *benchmarks* da aplicação de *Machine Learning* em processos químicos, uma vez que sua modelagem e simulação é muito difundida.

Dessa forma, nesse trabalho foram utilizadas técnicas de *Machine Learning*, mais especificamente as Redes Neurais Artificiais (*RNA's*) para estimar a concentração do produto de interesse no reator de Van de Vusse de maneira que a mesma funcione como um *Soft Sensor*. Os dados utilizados para a construção do *Soft Sensor*, foram obtidos a partir da modelagem e simulação do reator CSTR (*Continuous stirred tank reactor*) com reação de Van de Vusse.

2. OBJETIVOS

2.1 Geral

Foi desenvolvido e implementado, em linguagem *python*, uma estratégia de inferência estatística através de um *Soft Sensor* no processo de Van de Vusse utilizando modelos de *machine learning*, mais especificamente as RNA' s.

2.2 Específicos

- Foi modelado e simulado o processo de Van de Vusse através de balanços de massa e energia;
- Foi desenvolvida estratégias na forma de ferramenta computacional que estimam em tempo real variáveis do processo que não podem ser obtidas automaticamente, utilizando técnicas de *Machine Learning* supervisionado: Redes Neurais *Multilayer Perceptron (MLP)*;
- Foi avaliado o desempenho estatístico da estratégia desenvolvida através de índices apropriados de eficiência entre os dados obtidos pelo *Soft Sensor* e os dados obtidos pela simulação do processo.

3. REVISÃO BIBLIOGRÁFICA

De acordo com os objetivos apresentados neste trabalho, foi realizada uma revisão bibliográfica, abordando uma descrição da modelagem e simulação do reator de Van de Vusse (*benchmark*), as técnicas de *Machine Learning*, e, por fim, a linguagem de programação *Python* a ser utilizada no desenvolvimento do trabalho.

3.1 Reação de Van de Vusse

O mecanismo de reação apresentado por Van de Vusse se baseia na produção de ciclopentenol (B) a partir do ciclopentadieno (A) por meio da adição eletrofílica ácido catalisadora de água na solução diluída. São obtidos como subprodutos da reação o dicitlopentadieno (D) e o ciclopentanediol (C), como pode ser visto no esquema reacional abaixo (VAN DE VUSSE, 1964). O reator de Van de Vusse considerado nesse trabalho é um CSTR, ou seja, um reator contínuo perfeitamente agitado, de forma que não haja gradientes das suas variáveis em termos da localização espacial dentro dele. Neste reator ocorrem 3 reações: $A \rightarrow B \rightarrow C$ e $2A \rightarrow D$, cada uma com sua respectiva taxa cinéticas: r_1 , r_2 e r_3 .



Onde todos os k seguem a equação de Arrhenius:

$$k_i = k_{0i} * \exp \left(-\frac{E_i}{T} \right) \text{ com } i = 1,2,3 \quad (7)$$

Onde:

E_i : É a energia de ativação i.

T: É a temperatura em Kelvin.

r_i : É a taxa da reação.

C_a e C_b : São as concentrações de A e de B.

k_i : É a constante da reação.

k_{i0} : É o fator pré-exponencial.

As equações do modelo dinâmico partem da aplicação dos princípios fundamentais de conservação aos balanços de massa:

$$\frac{d}{dt}(V * C_a) = \text{entra} - \text{sai} + \text{gerado} - \text{consumido} \quad (8)$$

$$\frac{d}{dt}(V * C_a) = C_a * \frac{dV}{dt} + V * \frac{dC_a}{dt} \quad (9)$$

$$\frac{d}{dt}(C_a) = \frac{(F_{in} * C_{ain} - F_{out} * C_a)}{V} - k_1 * C_a - k_3 * C_a^2 - (C_a * \frac{dV}{dt})/V \quad (10)$$

$$\frac{d}{dt}(C_b) = \frac{(-F_{out} * C_b)}{V} + k_1 * C_a - k_2 * C_b - (C_b * \frac{dV}{dt})/V \quad (11)$$

$$\frac{d}{dt}(C_c) = \frac{(-F_{out} * C_c)}{V} + k_2 * C_b - (C_c * \frac{dV}{dt})/V \quad (12)$$

$$\frac{d}{dt}(C_d) = \frac{(-F_{out} * C_d)}{V} + k_3 * C_a^2 - (C_d * \frac{dV}{dt})/V \quad (13)$$

A variação de volume é baseada na equação do volume de tanque simples:

$$\frac{dV}{dt} = F_{in} - F_{out} = F_{in} - C_v * \sqrt{h_r} \quad (14)$$

F_{in} : É a vazão de entrada.

C_v : É a constante de válvula.

F_{out} : É a vazão de saída.

h_r : É o nível de líquido no reator.

A vazão de saída é proporcional à raiz quadrada do nível de líquido.

No balanço de energia, a fim de dissociar os efeitos da temperatura da água de resfriamento e a vazão, assumiu-se que a água de resfriamento é adicionada em um tanque de resfriamento perfeitamente misturado de volume muito inferior ao reator. Assumimos volume constante ao tanque de resfriamento e que em seu interior as propriedades da água são homogêneas, uma suposição válida uma vez que a vazão da água de resfriamento é muito superior ao volume do tanque (SOARES, 2017). Assim:

$$\frac{dT}{dt} = F_{in} * T_{in} - F_{out} * T)/V - (q_r - q_g)/(\rho * V * cp) - (T * dV/dt)/V \quad (15)$$

T_{in} : É a temperatura da corrente de entrada

q_r : É o calor removido pelo trocador de calor.

T: É a temperatura do reator.

V_r : Volume de reator.

Q_k : Taxa de refrigeração.

m_k : Massa do refrigerador

K_w : Coeficiente de transferência de calor.

Ar: Superfície do reator.

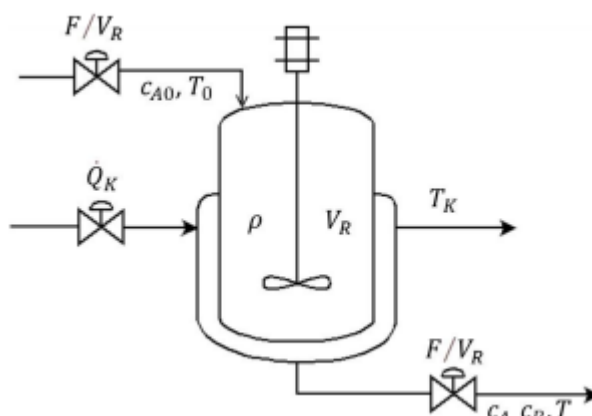
q_g : É o calor gerado pelas reações.

ρ : É a densidade.

C_p : É a capacitância térmica da mistura.

Em seguida, encontra-se a Figura com o esquema do reator para a reação de Van de Vusse:

Figura 1 – Esquema de reator



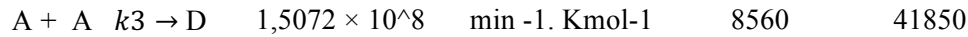
Fonte: KLATT & ENGELL (1998) - Adaptado

As equações do modelo do processo demonstradas são caracterizadas matematicamente como um sistema de EDO's (Equações Diferenciais Ordinárias) acopladas, frequentemente resolvidas, segundo a literatura, pelo método de Euler ou Runge Kutta.

Os parâmetros cinéticos da reação assim como as propriedades do reator/camisa são mostrados nas Tabelas 1 e 2.

Tabela 1 – Parâmetros cinéticos utilizados.

Reação	k_{i0}	Unidade k_{i0}	Energia de Ativação E_i (K)	Entalpia de Reação ΔHR (kJ/kmol)
A $k_1 \rightarrow B$	$2,145 \times 10^{10}$	min ⁻¹	9758,3	-4200
B $k_2 \rightarrow C$	$2,145 \times 10^{10}$	min ⁻¹	9758,3	11000



Fonte: KLATT & ENGELL (1998)

Tabela 2 – Propriedades do reator.

Parâmetro	Valor	Unidade
ρ	934,2	Kg/m ³
C_p	3,01	Kj/(Kg.k)
K_w	67,2	Kj/(m ² .min.k)
AR	0,215	m ²
VR	0,01	m ³
mK	5	Kg
C_pK	2	Kj/(Kg.k)

Fonte: KLATT & ENGELL (1998)

3.2 Inferência estatística e *Soft Sensor*

Inferência estatística é, em um contexto mais geral, estimar qual a distribuição probabilística de um certo dado. É usada para estimar o valor de certas variáveis em um determinado espaço de tempo. Dados do processo, como concentrações, demoram horas em análises laboratoriais para que se possa obter seu valor no momento em que a amostra foi retirada, além do alto custo dessas análises. Por conta disso o uso da inferência estatística é de suma importância nesses processos.

Nos últimos anos o estudo a respeito de *Soft Sensor* vem crescendo amplamente junto com sua utilização no controle de processos industriais para melhorar a qualidade do produto e garantir a segurança na produção. (YAN, et al 2004).

O desenvolvimento de modelos que sejam capazes de realizar a inferência dependeria do modelo da planta industrial, mas muitas vezes o modelo não está disponível devido à complexidade dos processos industriais. Como resultado, os modelos empíricos são os mais

populares para desenvolver *Soft Sensor*. O problema da modelagem empírica é encontrar um modelo com o melhor desempenho de generalização e previsão, com os dados empíricos. Atualmente, as técnicas de modelagem de *Soft Sensor* baseadas em modelos empíricos e incluem estatísticas multivariadas e *RNA's* (YAN, et al 2004).

3.3 Aprendizado de máquina *machine learning*

A busca pela resolução de problemas complexos vem provocando a procura pelo desenvolvimento de sistemas capazes de efetuar tal tarefa. Estes sistemas, que atualmente são o foco da Inteligência Artificial (IA), são compostos por algoritmos inteligentes que envolvem lógica, raciocínio dedutivo, sistemas especialistas e sistemas de *Machine Learning* (RUSSEL e NORVING, 2010).

Machine Learning é um ramo da IA que tem como uma das características o auto aprendizado a partir de exemplos, como aplicar modelagem empírica, detectar padrões e melhorar performance baseando-se unicamente em histórico de dados, reduzindo a necessidade de intervenção humana (FACELI, 2011).

Segundo, Soares (2017), existem 3 tipos de problemas típicos em *Machine Learning*, são eles:

- Aprendizado supervisionado é o que tem sido mais utilizado no mundo. Esse tipo de problema se baseia em relacionar uma entrada X com uma saída Y, a partir de dados fornecidos pelo usuário. A formulação de problemas nesse caso é bem direta, sem precisar de uma formalização matemática rígida ou de conhecimentos estatísticos profundos. É a preferida entre os engenheiros. Aprendizado supervisionado é utilizado também em problemas de classificação, onde a saída é um booleano (classes), e de regressão, em que a saída é um valor numérico. Problemas típicos são: detecção de falha (classificação), em que a saída é verdadeiro ou falso, 1 ou 0, para a pergunta, —há falha?; simulação da saída de um sistema é um problema típico de regressão (soft sensor), onde tenta mapear uma saída numérica Y a partir de uma entrada X.
- Aprendizado não supervisionado é quando apenas a entrada X é fornecida ao sistema. Entre os cientistas da computação é o tipo de aprendizado mais estudado, por ser mais próximo ao realizado pelo cérebro humano. Problemas típicos são redução de

dimensionalidade e remoção de ruídos. Ele exige uma formulação mais cuidadosa do problema, uma vez que o objetivo nem sempre é claro. Uma vantagem interessante do aprendizado não supervisionado é que ele pode ser automatizado a partir da captura de dados em tempo real. Em métodos supervisionados é necessário ter certeza de que um X corresponde a um Y, já que muitas vezes X e Y são avaliados por sistemas diferentes e analisar se o dado adquirido não é um ponto excepcional ou repetido. Se há apenas o X essa etapa manual é dispensada.

- Em aprendizado reforçado o programa interage com um sistema dinâmico e tenta alcançar um objetivo. Os famosos carros que se dirigem sozinhos são um tipo de aplicação de aprendizado reforçado. Aprendizado reforçado é uma generalização do campo da programação dinâmica, que é estudada a um tempo em processos químicos para um controle mais fino de processos variantes no tempo. (LEWIS & VRABIE, 2009; LEE & LEE, 2005).

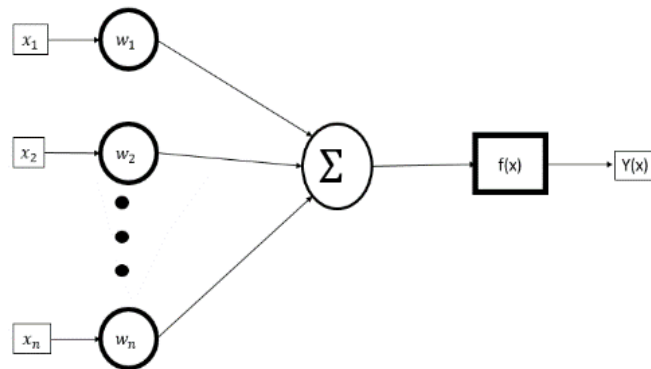
Para um treinamento eficiente, é necessário que os algoritmos utilizados sejam otimizados para garantir a convergência e velocidade nos resultados. E isso é especialmente verdade para as RNA's, que dependem muito da estimativa inicial dos pesos e de algoritmos que não se prendam a mínimos locais (LECUN, et al. 2012).

3.4 Redes neurais artificiais

A *Machine Learning* utilizada nesse trabalho, *RNA's*, assim como no cérebro, possui neurônios artificiais capazes de se “adaptar” e “aprender” com o ambiente. Esses tipos de modelos têm sido foco de atenção devido à ampla possibilidade de aplicação e disponibilidade para uso do conhecimento empírico armazenado. Usando algoritmos de aprendizado, elas podem reconhecer padrões ocultos e correlações em dados brutos, agrupá-los e classificá-los, e com o tempo aprender e melhorar continuamente. As *RNA's* se baseiam na forma de como o cérebro humano funciona, usando como base a conexão de neurônios artificiais. Para o uso desse modelo é necessário o uso de uma função de ativação que recebe uma soma ponderada de diversas entradas feitas por cada neurônio, estabelecendo pesos entre as conexões dos neurônios e tornando a RNA capaz de prever e de classificar dados (HAYKIN, 2009).

A Figura 2 ilustra a estrutura geral de um neurônio artificial, também conhecido como *perceptron*.

Figura 2 - Estrutura geral de um neurônio artificial



Fonte: Autor (2022)

Vemos que a estrutura recebe as entradas x_i , realiza uma soma ponderada com os pesos w_i , e segue para a função de ativação $f(x)$ para se obter a saída y . Matematicamente, esse processo é expresso pela equação 1, que representa a equação da estrutura *perceptron*:

$$y = f\left(\sum_{i=1}^n w_i x_i\right) \quad (16)$$

A Rede neural *perceptron* com apenas duas camadas que tem a estrutura de 3 neurônios na camada de entrada e 4 neurônios na camada de saída era um dos primeiros modelos estudados. Mas segundo Minsky e Papert (1988), entre os anos 50 e 60 a estrutura *perceptron* foi objeto de intensa pesquisa, no entanto em 1969 foi provado matematicamente que este tipo de estrutura de processamento apresenta limitações importantes e podem ser aplicadas com sucesso a uma classe muito restrita de problemas.

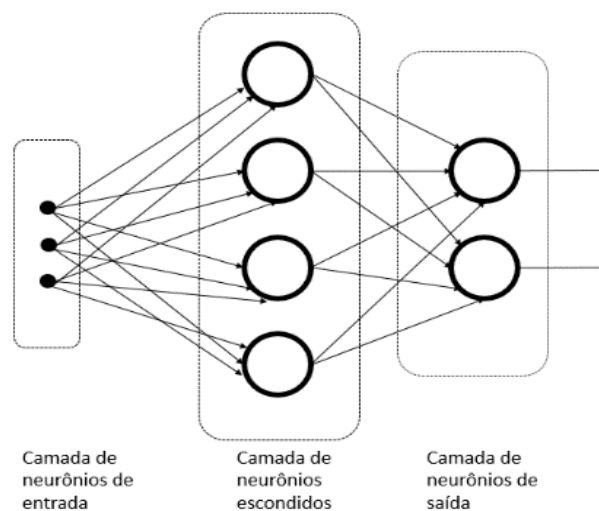
Segundo Iyoda (2000) foi provado que o *perceptron* é capaz de resolver apenas problemas linearmente separáveis. No entanto, com a utilização de redes de múltiplas camadas com pelo menos uma camada escondida (camada que não é nem entrada, nem saída), ou *perceptron* multicamadas (MLP – *multilayer perceptron*), muitas das limitações apresentadas pelo *perceptron* deixam de existir.

Além da estrutura *perceptron* conectados, existem inúmeras arquiteturas de redes neurais, como no caso da arquitetura *multicamadas* que é, sem dúvida, a mais frequentemente encontrada na literatura. Entre as razões para a sua popularidade pode-se citar sua capacidade de aproximação universal e sua flexibilidade para formar soluções de qualidade para uma ampla classe de problemas, a partir de um mesmo algoritmo de aprendizado (IYODA, 2000).

3.5 Redes neurais multicamadas

O *perceptron* multicamada (MLP – *Multilayer Perceptron*), são arquiteturas onde os neurônios estão dispostos em duas ou mais camadas de processamento, essas camadas se situam entre a camada de entrada e de saída, e conferem à rede uma ordem mais elevada em relação aos dados de entrada. Isso acontece devido ao maior número de conexões e conseqüentemente, maior dimensão das interações (BEDOR & FREITAS, 2020).

Figura 3 - rede com uma camada de entradas, uma camada escondida e uma camada de saída



Fonte: Autor (2022)

Por simplicidade, esta arquitetura é referida como uma rede 3-4-2, isto é, 3 neurônios de entrada, 4 neurônios escondidos e 2 neurônios de saída. Como outro exemplo, uma rede neural com p entradas, h_1 neurônios na primeira camada escondida, h_2 na segunda camada escondida e q neurônios na camada de saída é dita ser uma rede $p-h_1-h_2-q$ (IYODA, 2000).

3.6 Treinamento

A mais importante propriedade de uma rede neural artificial é sua capacidade de aprendizado. Aprendizado supervisionado é o mais comumente utilizado uma vez que é aplicado na elaboração de *Soft Sensor* e detecção de falhas.

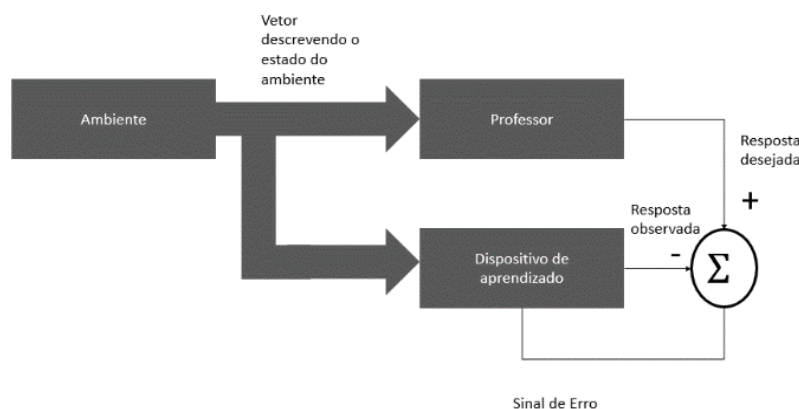
Não existe uma definição precisa universalmente aceita de “aprendizado”. No contexto de redes neurais artificiais, Haykin (1999) define aprendizado da seguinte forma: “Aprendizado é um processo pelo qual os parâmetros livres de uma rede neural são adaptados através de um processo de estímulo pelo ambiente no qual a rede está inserida”.

Segundo Iyoda (2000), o aprendizado supervisionado é caracterizado pela presença de um “professor” externo. A função do “professor” durante o processo de aprendizado é suprir a rede neural com uma resposta desejada a um determinado estímulo apresentado pelo ambiente. Definimos um sinal de erro como a diferença entre resposta desejada e a resposta observada na saída da rede neural. Os parâmetros da rede são então ajustados de acordo com o sinal de erro. Neste aprendizado uma rede neural aprende através de um processo iterativo de ajustes aplicados aos seus pesos sinápticos e limiares, o qual pode ser expresso na forma de um algoritmo computacional.

Faceli (2011) afirma que em tarefas de estimação de variáveis, o objetivo dos métodos de aprendizado supervisionado é encontrar uma função (também chamada de “modelo” ou soft sensor) a partir de um conjunto de exemplos (dados de treinamento) que possa ser utilizada para prever um valor de uma variável resposta Y , com base nos valores de seus atributos de entrada (variáveis preditoras X).

A Figura 4 apresenta um diagrama de blocos de um sistema com aprendizado supervisionado.

Figura 4 - Diagrama de blocos de um sistema com aprendizado supervisionado RNA



Fonte: Autor (2022)

3.7 Programação em python

A linguagem *Python* vem sendo amplamente utilizada desde 2019, afinal, por conta da sua versatilidade, esse tipo de código oferece muitas oportunidades no processo de desenvolvimento digital.

Para as empresas, a linguagem *Python* tem grande valor, pois além de ser gratuita é muito utilizada na ciência de dados e na automação de sistemas. Além disso, pode ser aplicada em projetos bastante distintos, como em aplicações web, softwares de reconhecimento facial e sensores virtuais.

A linguagem de programação *Python* vem se tornando cada vez mais utilizada na área de engenharia devido sua gratuidade em conjunto com um desenvolvimento crescente de novas bibliotecas, que conta com apoio de uma ampla comunidade. Com isso esse trabalho busca implementar diversas bibliotecas em *Python* para a construção do modelo de inferência para o Reator de Van de Vusse.

Algumas bibliotecas mais comuns a serem utilizadas no desenvolvimento do algoritmo do soft sensor são: Numpy, que serve para computação científica e cálculos de álgebra linear, a matplotlib, para a construção de gráficos e diagramas, a Dask e Pandas, para manipulação de dados, e a Scikit-learn, para aplicação de técnicas de aprendizado de máquina.

3.8 Computação em nuvem

A computação na nuvem ou *Cloud Computing* é um novo modelo de computação que permite ao usuário final acessar uma grande quantidade de aplicações e serviços em qualquer lugar e independente da plataforma, bastando para isso ter um terminal conectado à “nuvem” (SILVA, 2010).

A nuvem é representada pela internet, isto é, a infra-estrutura de comunicação composta por um conjunto de hardwares, softwares, interfaces, redes de telecomunicação, dispositivos de controle e de armazenamento que permitem a entrega da computação como serviço (PEDROSA et al., 2011).

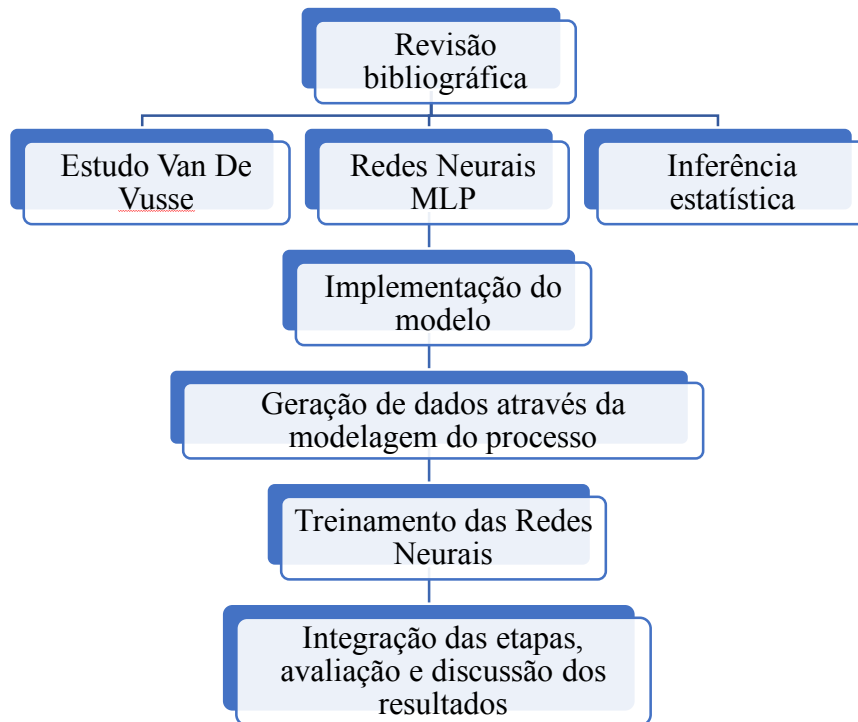
Outro ponto importante para o entendimento do modelo de computação na nuvem refere-se aos participantes da nuvem. Estes podem ser divididos em três grandes grupos: Provedor de serviço, Desenvolvedor e Usuário. O provedor é responsável pela tarefa de disponibilizar, gerenciar e monitorar toda a infra-estrutura da nuvem, garantindo o nível do serviço e a segurança adequada de dados e aplicações. Já o desenvolvedor deve ser capaz de prover serviços para o usuário final, a partir da infra-estrutura disponibilizada pelo provedor de serviço. Enquanto o usuário final é o consumidor que irá utilizar os recursos oferecidos pela nuvem computacional (SILVA, 2010).

Enfim, a computação na nuvem representa um novo modelo de serviço capaz de fornecer todo o tipo de processamento, infra-estrutura e armazenamento de dados através da internet (tanto como componentes separados ou uma plataforma completa) baseado na necessidade do usuário (PEDROSA et al., 2011).

Dessa forma, todo algoritmo foi desenvolvido na plataforma do Google Colab, permitindo a qualquer pessoa escrever e executar códigos Python arbitrário pelo navegador de maneira gratuita, com acesso a GPU's sem custo, o que foi especialmente adequado para aprendizado de máquina e análise de dados.

4. METODOLOGIA

A metodologia utilizada no trabalho pode ser visualizada na Figura 5 e consistiu inicialmente de uma revisão da literatura acerca do estudo da inferência estatística, do processo de Van de Vusse e das Redes Neurais *MLP*. Os dados para a avaliação da eficiência da *Machine Learning RNA's* como *Soft Sensor*, foram gerados através da modelagem do reator de Van de Vusse representado matematicamente a partir de seus balanços de massa e energia com os sistemas de equações diferenciais ordinárias resolvidas numericamente pelo método de Euler. Logo após, foi implementada a etapa do modelo e, em seguida, a integração das etapas junto com a avaliação dos resultados e discussões.

Figura 5 – Metodologia desenvolvida no trabalho

Fonte: Autor (2022)

A avaliação das redes neurais como modelo para identificação do reator de Van de Vusse, ou seja, o soft sensor, se deu por algumas etapas, onde a primeira consistiu-se em definir um caso base que serviu como referência para o treinamento da rede, avaliando os resultados. Para isso, foi realizada a definição dos seguintes parâmetros e hiperparâmetros:

- Seleção dos dados do estudo
- Quantidade de dados
- Quantidade de neurônios na camada escondida
- Quantidade de camadas escondidas

4.1 Seleção dos dados de estudo

Para o soft sensor foram consideradas como variáveis de saída a temperatura do reator (T) e a concentração de B na saída (C_B). Já as variáveis de entrada consistem na vazão da corrente de entrada do reator (F), no calor retirado do reator (Q_r), na concentração inicial de A (C_{A0}) e na temperatura da corrente de entrada (T_0). Foi adotado para o presente trabalho a vazão de 0,0157 m³/min e a concentração inicial de A em 5,1Kmol/m³.

Os limites de cada variável utilizados para gerar os sinais de entrada foram selecionados conforme a Tabela 3.

Tabela 3 - Definição dos limites de cada variável de entrada para a região de estudo

<i>Variáveis</i>	<i>Limite inferior</i>	<i>Limite superior</i>	<i>Unidade</i>
F_0	0.0015	0.03	m ³ /min
Q_r	-500	500	KJ/min
Ca_0	4	6	Kmol/m ³
T_0	373,15	423,15	K

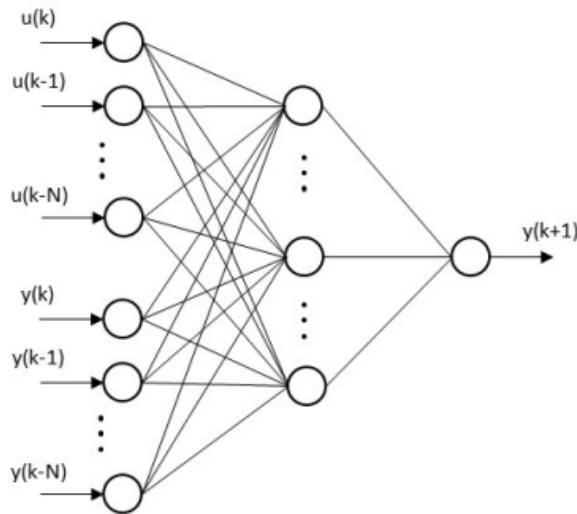
Fonte: Autor (2022)

4.2 Tratamento de dados

Em geral, a capacidade de modelar sistemas dinâmicos se dá pela utilização de dados atrasados no tempo, ou seja, uma RNA cujas entradas são compostas por valores passados das próprias variáveis de entrada e das variáveis de saída (SCHNITMAN, 1998).

Na Figura 6 é apresentada a estrutura geral de uma rede que recebe dados atrasados.

Figura 6 - Estrutura geral de rede com dados atrasados



Fonte: SILVA (2014)

Segundo Silva (2014), essa arquitetura pode ser utilizada para predição ao longo do tempo, onde a rede é treinada para prever apenas um passo adiante, que é realimentado a rede como entrada para o cálculo da próxima predição, os outros dados de entrada da rede são deslocados para trás e isto é feito iterativamente até que as predições tenham sido obtidas.

4.2.1 Normalização dos dados

Normalização é o processo de modelar o banco de dados projetando a forma como as informações serão armazenadas a fim de eliminar, ou pelo menos minimizar, a redundância no banco. Tal procedimento é feito a partir da identificação de uma anomalia em uma relação, decompondo-as em relações melhor estruturadas. O processo de normalização das variáveis é necessário para os modelos de RNA pelo fato de evitarem diferenças de magnitude entre os pesos, o que pode gerar viés no modelo e fazer com que sejam extraídos padrões não reais relacionados à natureza do problema (KARRI et al., 2018).

A normalização utilizada consiste em dimensionar as amostras individuais para se ter uma norma unitária, partindo da seguinte expressão:

$$X' = \frac{(X - X_{min})}{X_{max} - X_{min}} \quad (17)$$

X' : valor normalizado.

X_{\max} : Maior valor original.

X : valor original.

X_{\min} : Menor valor original.

4.3 Sinais de entrada

Para os sinais de entrada que foram utilizados como excitação do sistema será utilizado o sinal PRBS (Pseudorandom Binary Sequence), que é um sinal periódico que comuta entre dois níveis (Tabela 3), com tempo mínimo de permanência (T_b) em cada valor. Segundo Aguirre (2004), o tempo de amostragem pode ser definido como o período entre duas amostras t_s , de forma que t_s é o tempo ideal que nos forneça dados amostrados que preservem as características essenciais do dado original. Logo para esse trabalho, a termos de tempo de amostragem t_s , $3t_s \leq T_b \leq 5t_s$, então foi escolhido $T_b = 3t_s$ (AGUIRRE, 2014).

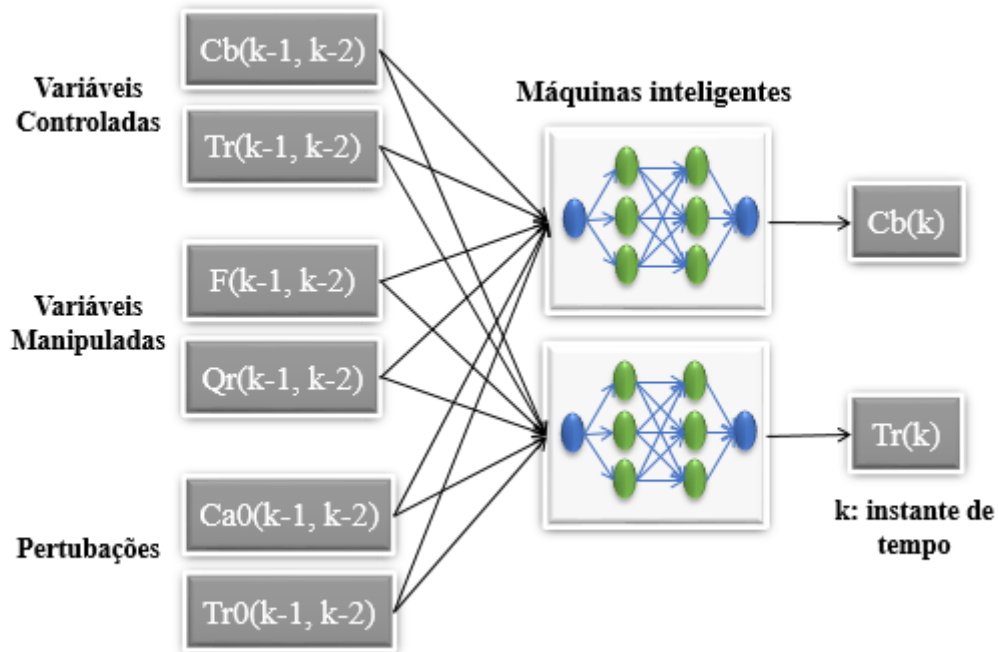
4.4 Detalhes do modelo utilizado

A estrutura da rede é composta de 12 entradas (Cb, Tr, F, Qr, Ca0, T0), com 2 dados atrasados, já a saída da rede é composta por duas variáveis, Cb e Tr, que serão preditas no tempo atual. SILVA (2014), utilizou o modelo de rede neural observado na Figura 7, em que utilizou 1 rede com duas saídas. Porém segundo a literatura é mais eficiente considerar 2 redes separadas, com saídas diferentes e mesmas variáveis de entrada, o chamado desacoplamento de rede neural (HAYKIN, 1999). Vale salientar que o conjunto de dados utilizados em ambas as redes, Figura 8 e Figura 9, foi o mesmo.

Para o treinamento foi utilizada a *Sklearn*, que se trata de uma biblioteca com várias funcionalidades e aplicações em *Machine Learning*, presente na linguagem *Python*.

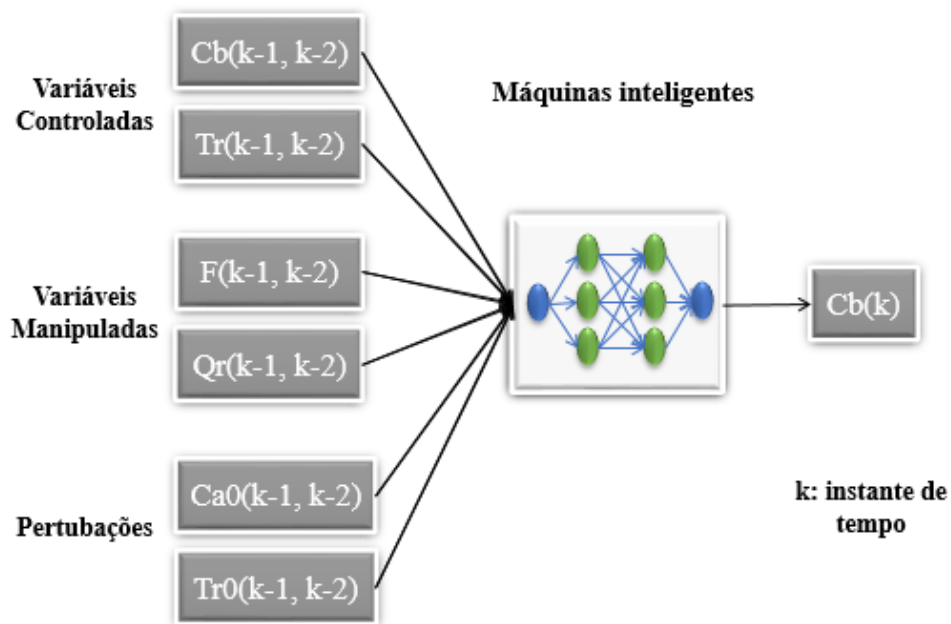
A Figura 7 mostra a estrutura da utilizada por Silva (2014).

Figura 7- Estrutura da rede utilizada por Silva



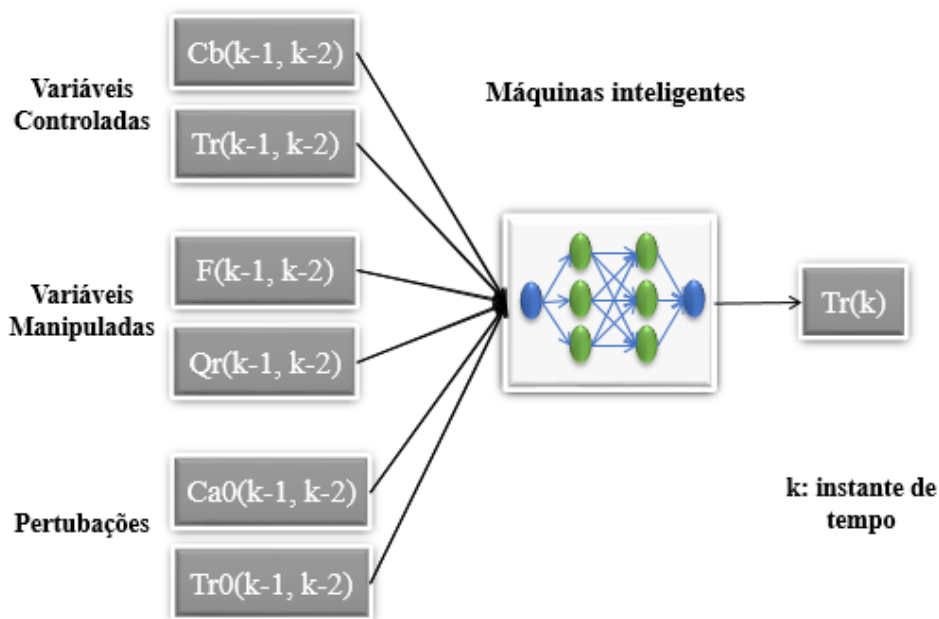
Fonte: SILVA (2014) - Adaptado

Figura 8 - Estrutura da rede utilizada Cb



Fonte: Autor (2022)

Figura 9 - Estrutura da rede utilizada Tr



Fonte: Autor (2022)

Os hiperparâmetros dos modelos de redes neurais, tais como, atraso de dados, tolerância, otimizador, número de neurônios nas camadas, foram obtidos pelo método heurístico de tentativa e erro.

Os dados foram otimizados pelo algoritmo de Adam, que é um otimizador baseado em gradiente estocástico, e que possui uma boa eficiência quando utilizado para trabalhar com uma grande quantidade de dados, já a número de iterações da rede (Épocas) foi de 300, para garantir a convergência dos dados seguindo a tolerância (Tol) pré-estabelecida.

A quantidade de neurônios da camada escondida pode ser definida de várias formas, mas deve-se tomar cuidado para que a quantidade de neurônios não seja muito grande, de modo que a rede possa memorizar os dados de treinamento (*overfitting*), que é o ajuste eficiente somente de dados de treinamento, e nem muito pequena, de tal forma que a rede gaste tempo em excesso buscando encontrar uma representação ótima (HAYKIN, 2009).

Na tabela 4, temos os hiperparâmetros utilizados na rede.

Tabela 4 - Hiperparâmetros da rede MLP

Dados atrasados	Tol	Otimizador	Épocas	Número de neurônios nas camadas		
				Entrada	Escondida	Saída
2	0.000001	Adam	300	12	7	1

Fonte: Autor (2022)

Para isso foi utilizada uma prática bastante adotada, onde o número de neurônios na camada escondida é:

$$\text{N}^\circ \text{ Neurônios da camada escondida} = \frac{(\text{N}^\circ \text{ Entrada}) + (\text{N}^\circ \text{ Saída})}{2}$$

4.5 Critérios de avaliação

O modelo será válido caso tenha a capacidade de incorporar características essenciais dos fenômenos vitais para sua aplicação. Logo, para medidas de acurácia de predição foram utilizados a raiz do erro quadrático médio (RMSE) e o coeficiente de determinação R^2 , demonstrados pelas equações 18 e 19.

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2} \quad (18)$$

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \quad (19)$$

Onde n é o número de dados, y_i é o valor observado, \bar{y}_i é a média das observações, e \hat{y}_i é valor estimado ou previsão de y_i .

4.6 Ambiente de treinamento

Este trabalho foi desenvolvido utilizando um computador com processador AMD Quad-Core Processor A10-9600P, com HDD de 500 GB e 12 GB de memória RAM. Deve ser observado que pelo grande número de dados utilizados neste trabalho, da ordem de 10000 dados, computadores convencionais com até 16GB de memória RAM não são

suficientes para tratar problemas dessa magnitude, fazendo com que a computação em nuvem tenha se tornado a única opção utilizada no presente trabalho.

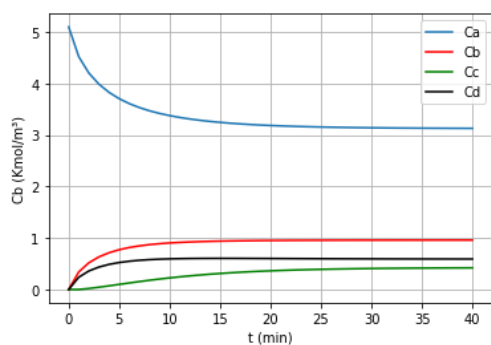
5. RESULTADOS E DISCUSSÕES

Os resultados deste trabalho são apresentados e discutidos de forma simultâneas, e estão organizados conforme a ordem de execução, formando uma linha de raciocínio e considerações interdependentes.

5.1 Modelagem do reator de Van de Vusse

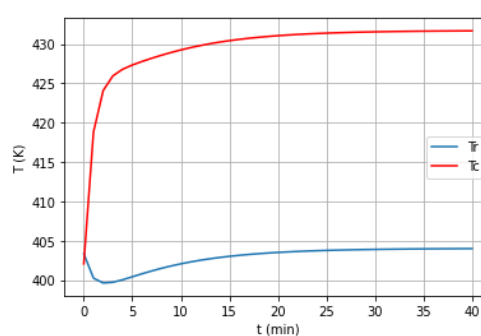
Através da modelagem e simulação que pode ser exemplificada nas equações do tópico 3.1, foi obtido os dados para elaboração do soft sensor. A simulação do processo durante o tempo de 40 minutos, para fins de estabilidade e melhor visualização, pode ser vista nas Figuras 10 e 11.

Figura 10 - Resposta das concentrações com a simulação



Fonte: Autor (2022)

Figura 11 - Resposta das temperaturas com a simulação



Fonte: Autor (2022)

Observa-se a estabilidade do modelo, uma vez que a partir dos dados iniciais de F , Q_r , Ca_0 e T_0 , o estado estacionário foi atingido.

Tabela 5 – Valores iniciais.

<i>Variáveis</i>	<i>Valores iniciais</i>	<i>Unidade</i>
F	0.0157	m ³ /min
Qr	400	KJ/min
Ca0	5.1	Kmol/m ³
T0	403.34	K

Fonte: Autor (2022)

5.2 Definição do tempo de amostragem

Para encontrar a menor constante de tempo do sistema, foram realizadas simulações dinâmicas do processo. Para isso foi feita a integração das equações diferenciais do modelo do reator de Van de Vusse, utilizando o método de Euler e obtendo as respostas das variáveis C_B e T em relação a funções degrau para todas as variáveis de entrada. As amplitudes dos degraus são exibidas na Tabela 6.

Tabela 6 - Amplitudes dos degraus aplicados às variáveis de entrada.

<i>Variáveis</i>	<i>Amplitude do degrau</i>	<i>Unidade</i>
F	0.01	m ³ /min
Qr	20	KJ/min
Ca0	-1,1	Kmol/m ³
T0	20	K

Fonte: Autor (2022)

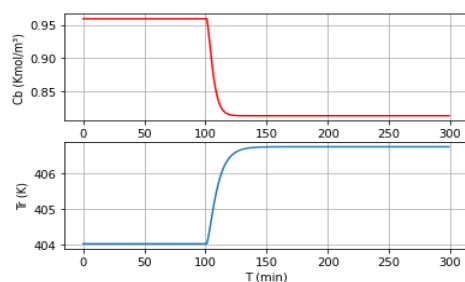
Para determinação da resposta com dinâmica mais rápida foi feita uma avaliação visual. Em seguida, foi calculado a constante de tempo a partir do método gráfico,

sugerido por SEBORG (2011). Para uma resposta ao degrau na entrada com comportamento aproximado a um modelo de 1º ordem, a resposta alcança 63,2% do seu valor final quando $t = \tau$, sendo τ a constante de tempo do processo (SEBORG, 2011).

Será utilizado como base, a menor constante de tempo do sistema para definição do tempo de amostragem do sinal superamostrado. Pois a constante de tempo do sistema é diretamente responsável pela quantidade de informações sobre a dinâmica do sistema nos dados. Onde, se o tempo de amostragem for muito maior do que a constante de tempo do sistema, poderá haver pouca informação sobre a dinâmica do sistema e se for muito menor, poderá ser pouco informativo e gerar ruído. Logo, de forma que o sinal superamostrado seja pequeno o suficiente, o valor encontrado para a menor constante de tempo será dividido por 100.

Para definir o tempo de amostragem, foram gerados os gráficos que mostram as respostas de C_b e T_r para os degraus de entrada em Ca_0 , T_0 , F e Q_r com as amplitudes exibidas na Tabela 4.

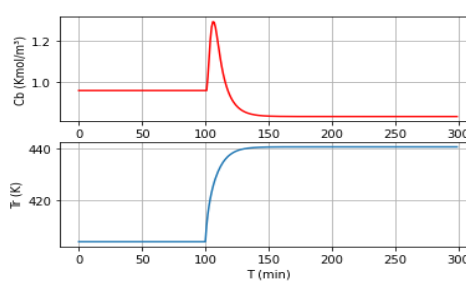
Figura 12 - Resposta ao degrau de Ca_0



$$C_b = 0.8124200899421974$$

$$T_r = 406.75771668329395$$

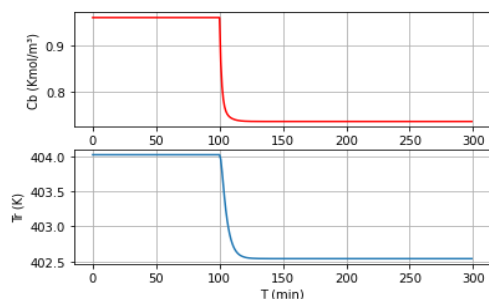
Figura 13 - Resposta ao degrau de T_0



$$C_b = 0.8338696175458776$$

$$T_r = 440.7681132547703$$

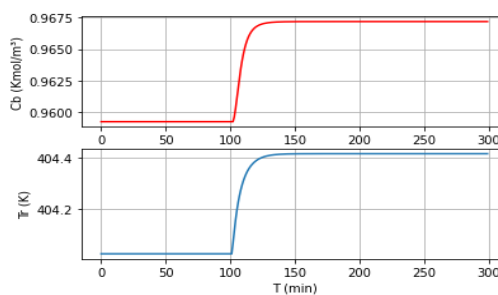
Figura 14 - Resposta ao degrau de F



$$C_b = 0.7361330248351169$$

$$T_r = 402.53895332360815$$

Figura 15 - Resposta ao degrau de Q_r



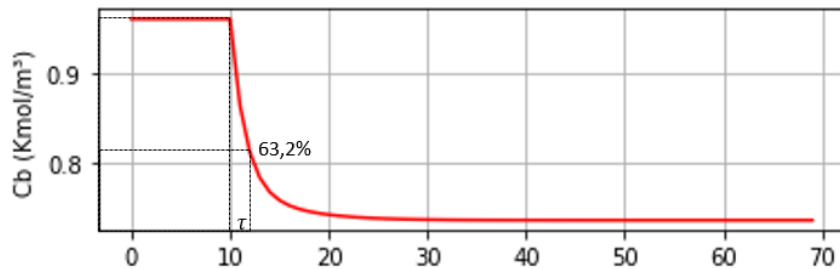
$$C_b = 0.967191369734291$$

$$T_r = 404.41407799500604$$

Fonte: Autor (2022)

A partir de uma análise visual, foi identificado que a resposta mais rápida foi a de C_b em relação ao degrau em F . Com isso calcula-se a constante de tempo através do método gráfico, como sugerido por Seborg (2011).

Figura 16 - Resposta ao degrau de F



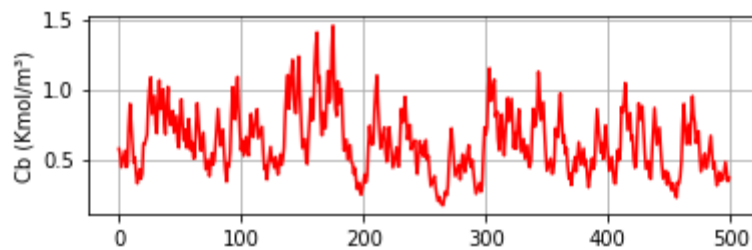
Fonte: Autor (2022)

O τ obtido foi de 2, com isso calcula-se o tempo de amostragem:

$$t_s = \frac{\tau}{100} = 0,02$$

Esse t_s foi utilizado para gerar o sinal de entrada de F como um PRBS, como parâmetro $T_b = 3t_s$, como critério de exemplificação, foram utilizados 500 dados para confecção da Figura 17.

Figura 17 - Exemplo da resposta de C_b em relação ao sinal de entrada PRBS

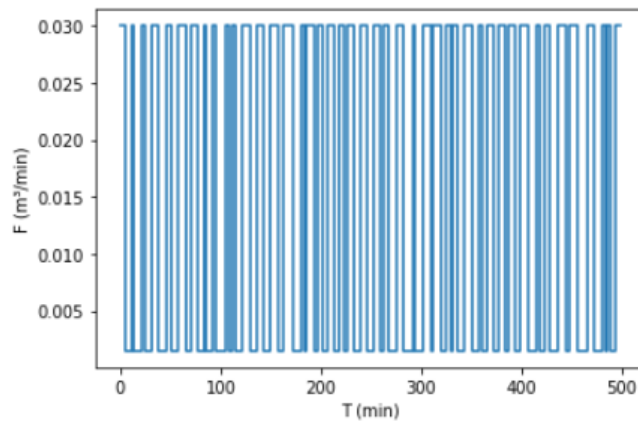


Fonte: Autor (2022)

5.3 Geração do sinal de entrada

O sinal de entrada PRBS foi gerado com base na entrada de F, a Figura abaixo ilustra o comportamento do sinal.

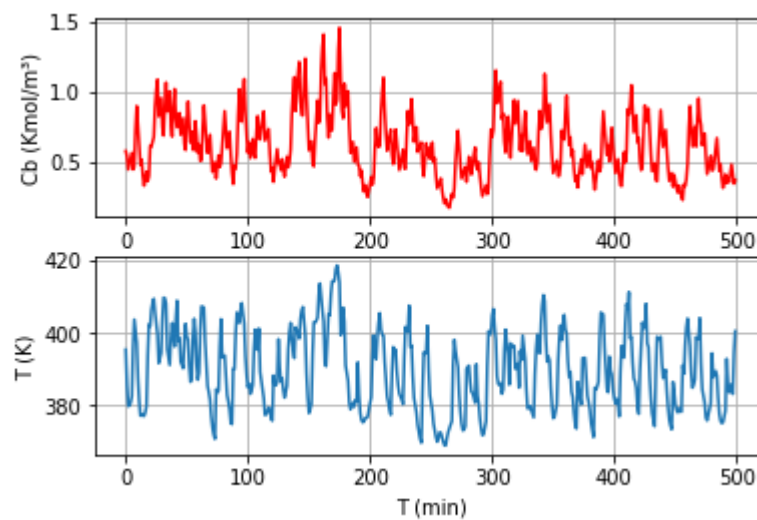
Figura 18 - Sinal de entrada PRBS de F em m³/min



Fonte: Autor (2022)

A seguir é mostrado a resposta do sistema para as variáveis de saída C_b e T_r utilizando o sinal PRBS em todas as variáveis de entrada conforme a tabela 4. Gerando dessa forma o conjunto de dados num total de N = 10.000.

Figura 19 - Resposta do sistema para o sinal PRBS



Fonte: Autor (2022)

5.4 Validação do modelo

Para a validação do modelo, os conjuntos foram separados em conjunto de treinamento e conjunto de teste, onde 80% dos dados foram para o conjunto de treinamento e o restante para o conjunto de teste. Os dados de treinamento foram utilizados para o treinamento da rede, após isso o sistema utiliza o conjunto de teste para estimar a taxa de erro. Como métrica de avaliação foram utilizados o *Score* e o RMSE, vale salientar que foram realizados vários testes onde os resultados podem ser vistos na tabela 7 e nas Figuras 20 e 21.

Tabela 7 - Dados de RMSE e Score.

Cb		Tr	
<i>Score</i>	RMSE	<i>Score</i>	RMSE
0,978	0,11	0,977	0,1159

Fonte: Autor (2022)

Vale salientar que os valores ótimos para o *Score* são próximos de 1 e os valores ótimos para o RMSE são próximo de zero.

Figura 20 - Dados de teste e de treino simulados versus dados preditos de Tr

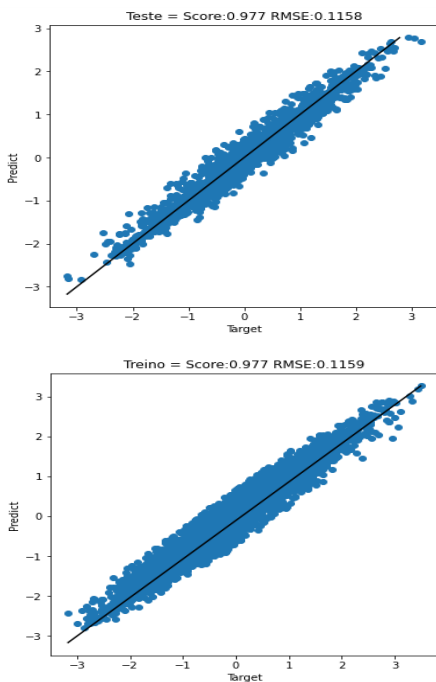
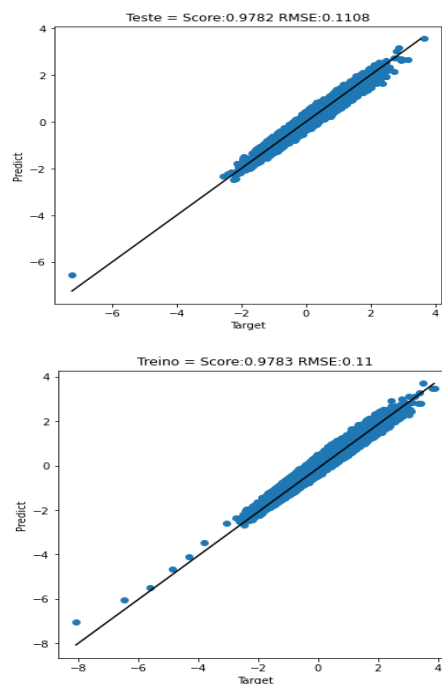


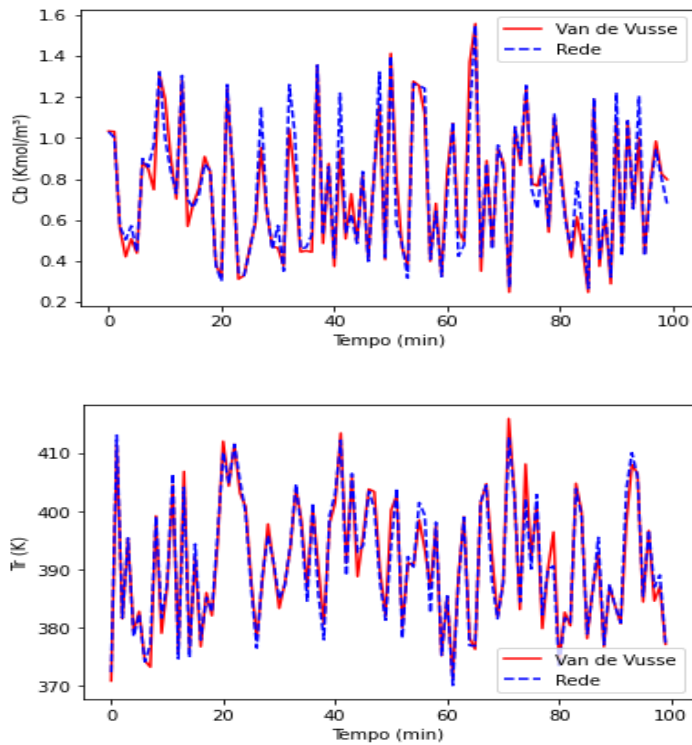
Figura 21 - Dados de teste e treino simulados versus dados preditos de Cb



Fonte: Autor (2022)

Na Figura 22 mostra o gráfico comparando a resposta obtida pelo modelo com os valores simulados do reator de Van de Vusse.

Figura 22 - Dados simulados versus comparados com os dados preditos



Fonte: Autor (2022)

Pela Figura 22 nota-se que o soft sensor teve um bom ajuste de dados de forma geral, obtendo uma boa previsão do comportamento de oscilatório do processo diante das perturbações. Para obtenção dos resultados foram necessárias 300 épocas na *RNA*.

6. CONCLUSÃO

O presente trabalho buscou como objetivo estudar e desenvolver estratégias para a utilização de Redes Neurais Artificiais como *Soft Sensor*, realizando a inferências dos dados no reator de Van de Vusse, que é um processo de reações complexas e conhecido por suas não linearidades.

O *Soft Sensor* efetuou uma boa inferência estatística do reator de Van de Vusse, onde demonstraram ser capazes de aprender as não linearidades do reator e suas propriedades, que é o maior desafio ao controle do reator.

Os resultados obtidos pelas redes foram estáveis, podendo ser usadas para predição de longos períodos de tempo e não apenas em um instante seguinte. Isso indica que os modelos de *Machine Learning* se mostram eficientes e podem ser usados com boa robustez como sensores virtuais. Sendo uma ótima opção em casos onde o custo de análises do processo é muito alto ou demorado, principalmente por sua rápida convergência, sem preocupação com a instabilidade do sistema na utilização após o treinamento e aprendizagem.

No geral, as técnicas baseadas em *Machine Learning* se mostram promissoras e preparadas para emprego na engenharia de processos químicos, principalmente com o advento das novas tecnologias, em engenharia química, que buscam criar computadores mais potentes e eficientes para baterem de frente com o esforço analítico e computacional necessário para resultados superiores.

7. REFERÊNCIAS

- AGUIRRE, L. A. Introdução à Identificação de Sistemas. Técnicas Lineares e Não-Lineares Aplicadas a Sistemas Reais. 2a edição ed. Belo Horizonte: Editora UFMG, 2004.
- BAUGHMAN, D. R. & LIU. Neural Networks in Bioprocessing and Chemical Engineering. Academic Press 1995.
- BEDOR, N. B. A. FREITAS, G. C. Estudo de identificação de um reator de Van de Vusse com redes neurais feedforward. Rio de Janeiro/EQ,2020. vii, 44 p. il
- DE SOUZA JUNIOR, M. B. Redes neurais multicamadas aplicadas a modelagem e controle de processos químicos, Tese de doutorado PEQ/COPPE/UFRJ, 1993.
- FACELI, K. LORENA, A. C. GAMA, J. CARVALHO, A. C. P. L. F. Inteligência Artificial: Uma Abordagem de Aprendizado de Máquina. 2011. LTC. Rio de Janeiro, RJ, 2011.
- GONÇALVES, G. A. A. ALVARISTO, E. L. SILVA, E. L. SOUZA JR, M. B. SECCHI, A. R. A comparison of performance and implementation characteristics of nmpc formulations, XX Congresso Brasileiro de Engenharia Química, Blucher Chemical Engineering Proceedings, Volume 1, 2015.
- HASTIE, T; TIBSHIRANI, R. FRIEDMAN, J., The elements of statistical learning, 20 Edition, 2008.
- HAYKIN, S. Neural Networks: A Comprehensive Foundation. 2nd Edition, Prentice- Hall, Englewood Cliffs, NJ, 1999.
- HAYKIN, S. S. Neural networks and learning machines. 3rd ed ed. New York: Prentice Hall, 2009.
- IYODA, E. M. Inteligência computacional no projeto automático de redes neurais híbridas e redes neurofuzzy heterogêneas. Campinas, SP, 2000.
- KLATT, K.-U. ENGELL, S. Gain-scheduling trajectory control of a continuous stirred tank reactor. Computers chem. Engng, v. 22, p. 491–502, 1998.
- LECUN, Y. BOTTOU, L. ORR, G. B. MÜLLER, K.-R. Efficient BackProp. In: G. Montavon, G. B. Orr, and K.-R. Müller (eds.) Neural Networks: Tricks of the Trade, 2nd ed. Springer LNCS 7700, 659-686, 2012.

- LEE, J. M. LEE. J, H.; Approximate dynamic programming-based approaches for input– output data-driven control of nonlinear processes. *Automatica* 41 (2005) 1281 – 1288, 2005.
- LEWIS, F, L; VRABIE, D; Reinforcement Learning and Adaptive Dynamic Programming for Feedback Control *IEEE CIRCUITS AND SYSTEMS MAGAZINE*. 2009.
- M. A. Minsky and S. A. Papen, *Perceptrons*-Expanded Edition. Cambridge, MA: MIT Press, 1988.
- MANNILA, H., Data mining: machine learning, statistics, and databases.. Int'l Conf. Scientific and Statistical Database Management. IEEE Computer Society. 1996.
- MINSKY, M. & PAPERT, S. *Perceptrons: an Introduction to Computational Geometry*, edição expandida, MIT Press, 1988.
- MONTAVON, G; GENEVIÉVE B., O; MÜLLER, K.R. et al. *Neural Networks: Tricks of the Trade..* 20 edition, 2012. DOI 10.1007/978-3-642-35289-8, 2012.
- PEDROSA, PAULO H. C. NOGUEIRA, T. *COMPUTAÇÃO EM NUVEM*. UNICAMP-SP, 2011.
- RUSSEL, S. NORVING, P. *Artificial Intelligence – A Modern Approach*. 3a edição, Prentice Hall, 2010.
- SCHNITMAN, L. Controladores preditivos baseados em redes neurais artificiais. Dissertação (Mestrado em Engenharia Elétrica) – Universidade Federal da Bahia, Salvador, 1998.
- SEBORG, D. E. *Process dynamics and control*. 3rd ed ed. Hoboken, N.J: John Wiley & Sons, Inc, 2011.
- SILVA, F. H. R. Um estudo sobre os benefícios e os riscos de segurança na utilização de Cloud Computing; 15f. Artigo científico de conclusão de curso apresentado no Centro Universitário Augusto Motta, UNISUAM-RJ, 2010.
- SILVA, G. DE C. Estudo de um Controlador Preditivo Não Linear Multivariável Baseado em Redes Neurais. Mestrado em Ciências—[s.l.] UFRJ, 2014.
- SOARES, Felipe Doval Rojas. Técnicas de *Machine Learning* Aplicadas a Inferência e Detecção e Diagnóstico de Falhas de Processos Químicos Industriais em Contexto Big Data - 2017 XVIII, 90 f, il. UFRJ, Escola de Química, Rio de Janeiro, 2017.

VAN DE VUSSE, J. G. Plug-flow Type Reactor Versus Tank Reactor. *Chemical Engineering Science*, v. 19, p. 964-997, 1964.

YAN, W., SHAO, H., & WANG, X. Soft sensing modeling based on support vector machine and Bayesian model selection. *Computers & Chemical Engineering*, 28(8), 1489–1498, 2004.