

UNIVERSIDADE FEDERAL DE ALAGOAS - UFAL
CENTRO DE TECNOLOGIA - CTEC
CURSO DE ENGENHARIA QUÍMICA

VITOR MARCEL DE MENDONÇA MONTEIRO

***DEEP LEARNING AUTOENCODER NO DESENVOLVIMENTO DE SOFT-SENSOR
PARA PROCESSOS DE FERMENTAÇÃO ALCOÓLICA EM BATELADA***

Maceió - AL
2025

VITOR MARCEL DE MENDONÇA MONTEIRO

***DEEP LEARNING AUTOENCODER NO DESENVOLVIMENTO DE SOFT-SENSOR
PARA PROCESSOS DE FERMENTAÇÃO ALCOÓLICA EM BATELADA***

Trabalho de conclusão de curso apresentado
como requisito parcial para obtenção do título
de Bacharel em Engenharia Química pela Uni-
versidade Federal de Alagoas.

Orientador/a: Prof. Dr. Jorge José de Brito Silva

Coorientador/a: Prof. Dr. Frede de Oliveira
Carvalho

**Catalogação na Fonte
Universidade Federal de Alagoas
Biblioteca Central
Divisão de Tratamento Técnico**

Bibliotecário: Marcelino de Carvalho Freitas Neto – CRB-4 – 1767

M775d Monteiro, Vitor Marcel de Mendonça.
Deep learning autoencoder no desenvolvimento de *soft-sensor* para processos de fermentação alcoólica em batelada / Vitor Marcel de Mendonça Monteiro. – 2025.
45 f. il. : figs. ; tabs. color.

Orientador: Jorge José de Brito Silva.
Co-orientador: Frede de Oliveira Carvalho.
Monografia (Trabalho de Conclusão de Curso em Engenharia Química).
Universidade Federal de Alagoas. Centro de Tecnologia. Maceió, 2025.

Bibliografia: f. 43-45.

1. Fermentação. 2. Redes neurais. 3. Soft-Sensor. 4. Autoencoder (Rede neural). I. Título.

CDU: 663:004.72

Folha de Aprovação

VITOR MARCEL DE MENDONÇA MONTEIRO

DEEP LEARNING AUTOENCODER NO DESENVOLVIMENTO DE SOFT-SENSOR PARA PROCESSOS DE FERMENTAÇÃO ALCOÓLICA EM BATELADA

Trabalho de conclusão de curso apresentado
como requisito parcial para obtenção do título
de Bacharel em Engenharia Química pela
Universidade Federal de Alagoas.

Banca Examinadora:

Orientador: Prof. Dr. Jorge José de Brito Silva
(Universidade Federal de Alagoas)

Prof. Dr. João Batista Maia Rocha Neto
(Universidade Federal de Alagoas)

Prof.^a Dra. Soraya Lira Alencar
(Universidade Federal de Alagoas)

AGRADECIMENTOS

Gostaria de agradecer a todas as pessoas que me incentivaram e ajudaram durante a graduação, em especial a Julia Evelyn, por me apoiar e estar comigo desde o início dessa jornada.

Agradeço imensamente aos amigos que fiz durante esse processo e que me deram suporte nos momentos mais difíceis, principalmente Lucas Carvalho, Amanda Santos, Miquéias Jacinto e Layla Barreto.

Agradeço aos professores Jorge Silva e Frede Carvalho. Sem eles, esse trabalho não poderia ter sido realizado. Sou grato por todos os conselhos recebidos, tanto acadêmicos quanto para a vida profissional, que me ajudaram a amadurecer e tomar decisões melhores.

Só é útil o conhecimento que nos faz melhores.

Sócrates

RESUMO

Fermentação alcoólica é um processo bioquímico no qual há conversão de açúcares simples em compostos de interesse comercial, como, por exemplo, o etanol. A indústria brasileira de etanol chegou a produzir 35,6 bilhões de litros na safra de 2019/2020. O processo produtivo é influenciado por diversas variáveis, tais como o tipo de levedura, contaminação bacteriana, concentração de açúcares no meio reacional, tempo de alimentação das dornas, temperatura, pH, entre outras, e isso torna o controle e otimização do processo complexo, principalmente quando essas variáveis devem ser medidas diretamente por sensores físicos convencionais ou por análises em laboratório. Ao contrário das medições convencionais, os *soft-sensors* utilizam algoritmos matemáticos, técnicas de inteligência artificial ou modelos estatísticos para inferir ou prever essas variáveis com base em informações indiretas, como medidas disponíveis, histórico de dados ou correlações entre variáveis no qual modelos clássicos não conseguem explicar. Processos em batelada, diferentemente de processos contínuos, apresentam dificuldades em relação à utilização de técnicas para elaboração de modelos empíricos para inferência, que é como a variável tempo é considerada e incorporada ao modelo. *Autoencoder* (AE) é uma rede de aprendizado profundo não supervisionado composta por um codificador e um decodificador que pode ser utilizada para reduzir a dimensionalidade dos dados e facilitar o treinamento de um modelo, a mesma vem sendo muito utilizada como classificadora na detecção e diagnóstico de falha, porém a sua utilização como *soft-sensor* ainda é relativamente nova. Com base nisso, o trabalho teve como objetivo explorar a utilização do AE, assim como algumas técnicas já bastante utilizadas na elaboração dos *soft-sensors* como as redes neurais artificiais (RNAs) e as *long short term memory* (LSTM) focando a problemática da incorporação da variável tempo aos modelos. Todas as três redes apresentaram bons resultados para prever a concentração de etanol, com valores de R^2 acima de 0,9 e erros abaixo de 2,5, entretanto a rede LSTM se destacou como a mais eficaz pois apresentou os melhores valores nas métricas avaliadas ($R^2 = 0,9973$, $RMSE = 0,4336$ e $MAE = 0,3055$). Já o modelo LSTM-*Autoencoder* mostrou-se uma alternativa promissora, capaz de manter boa precisão mesmo utilizando uma quantidade menor de informação ($R^2 = 0,9551$, $RMSE = 1,7722$ e $MAE = 1,5142$), se mostrando relevante para aplicações onde a eficiência computacional é um ponto crítico.

Palavras-Chave: Processo Fermentativo; Redes Neurais Artificiais; *Soft-Sensor*; *Autoencoder*.

ABSTRACT

Alcoholic fermentation is a biochemical process in which simple sugars are converted into commercially interesting compounds, such as ethanol, for example. The Brazilian ethanol industry reached a production of 35.6 billion liters in the 2019/2020 harvest. The production process is influenced by various variables, such as the type of yeast, bacterial contamination, sugar concentration in the reaction medium, feeding time of the fermenters, temperature, pH, among others, and this makes the control and optimization of the process complex, especially when these variables must be measured directly by conventional physical sensors or laboratory analyses. Unlike conventional measurements, soft sensors use mathematical algorithms, artificial intelligence techniques, or statistical models to infer or predict these variables based on indirect information, such as available measurements, historical data, or correlations between variables that classical models cannot explain. Batch processes, unlike continuous processes, present a difficulty regarding the use of techniques for the development of empirical models for inference, which is how the time variable is considered and incorporated into the model. An Autoencoder (AE) is an unsupervised deep learning network composed of an encoder and a decoder that can be used to reduce data dimensionality and facilitate model training. It has been widely used as a classifier in fault detection and diagnosis; however, its use as a soft-sensor is still relatively new. Based on this, the work aimed to explore the use of AE, as well as some techniques already widely used in the development of soft sensors such as artificial neural networks (ANNs) and long short-term memory (LSTM), focusing on the issue of incorporating the time variable into the models. All three networks showed good results in predicting ethanol concentration, with R^2 values above 0,9 and errors below 2,5. However, the LSTM network stood out as the most effective, presenting better metrics for the evaluated variables ($R^2 = 0.9973$, RMSE = 0.4336 and MAE = 0.3055). The LSTM-Autoencoder model, on the other hand, proved to be a promising alternative, capable of maintaining good accuracy even when using a smaller amount of information ($R^2 = 0.9551$, RMSE = 1.7722 and MAE = 1.5142), making it relevant for applications where computational efficiency is a critical point.

Keywords: Fermentation Process; Artificial Neural Networks; Soft-Sensor; Autoencoder.

LISTA DE FIGURAS

Figura 1 - Produção de cana-de-açúcar e etanol entre o período de 2019 e 2024. A informação para a safra 24/25 é uma estimativa.....	16
Figura 2 - Representação de um processo fermentativo em batelada alimentada.....	19
Figura 3 - Modelo adaptado de um <i>soft-sensor</i> com um <i>hardware</i> e um estimador.....	21
Figura 4 - Exemplos de aplicação em função do tipo de aprendizado.....	22
Figura 5 - Neurônio artificial e seus elementos básicos.....	23
Figura 6 - Representação de uma rede neural MLP de 3 camadas.....	25
Figura 7 - Rede Neural Recorrente Simples.....	25
Figura 8 - Arquitetura típica de um bloco LSTM. A rede é composta por um portão de entrada (<i>input gate</i>), um portão de esquecimento (<i>forget gate</i>), a célula (<i>cell</i>) e o portão de saída (<i>output gate</i>).....	26
Figura 9 - Estrutura básica de um <i>autoencoder</i>	27
Figura 10 - Fluxo da metodologia do trabalho.....	28
Figura 11 - Exemplos visuais de ajustes aos dados. A figura mais à esquerda representa uma falta de ajuste enquanto que a mais à direita representa o sobreajuste. A figura central representa um bom ajuste.....	32
Figura 12 - Gráficos das simulações feitas para geração e coleta de dados do processo.....	34
Figura 13 - Gráfico da simulação para geração de dados para o teste dos modelos.....	35
Figura 14 - Curvas de perda do treinamento e da validação. A figura (a) é a rede MLP, a (b) da rede LSTM e a (c) é do LSTM-AE.....	35
Figura 15 - Gráfico das concentrações de etanol real versus previsto pela MLP para o primeiro teste.....	36
Figura 16 - Gráfico das concentrações de etanol real versus previsto pela MLP para o segundo teste.....	37
Figura 17 - Gráfico de paridade entre os valores reais e previstos da MLP.....	37
Figura 18 - Gráfico das concentrações de etanol real versus previsto pela LSTM para o primeiro teste.....	38
Figura 19 - Variáveis do processo ao longo da simulação.....	39
Figura 20 - Gráfico de paridade entre os valores reais e previstos da LSTM para o segundo teste.....	39
Figura 21 - Gráfico das concentrações de etanol real versus previsto pela LSTM para o segundo teste.....	39

Figura 22 - Gráfico das concentrações de etanol real versus previsto pela LSTM-AE para o segundo teste.....	40
Figura 23 - Gráfico de paridade entre os valores reais e previstos da LSTM-AE para o segundo teste.....	41

LISTA DE TABELAS

Tabela 1 - Modelos cinéticos.....	20
Tabela 2 - Exemplos de aplicação de <i>soft-sensors</i> em processos químicos.	21
Tabela 3 - Funções de ativação e suas aplicações.	24
Tabela 4 - Valores das variáveis em cada simulação.	29
Tabela 5 - Valores das variáveis do processo.....	29
Tabela 6 - Valores de uma variável arbitrária A ao longo do tempo.....	30
Tabela 7 - Exemplo de como os dados são organizados e inseridos na rede.	30
Tabela 8 – Variáveis alimentadas em cada teste. Os subscritos 1, 2 e 3 representam o atraso da variável.	31
Tabela 9 – Parâmetros das redes.....	32
Tabela 10 – Métricas de desempenho da MLP para o primeiro teste.....	36
Tabela 11 - Métricas de desempenho da MLP para o segundo teste.....	37
Tabela 12 - Métricas de desempenho da LSTM para o primeiro teste.....	38
Tabela 13 - Métricas de desempenho da LSTM para o segundo teste.	38
Tabela 14 - Métricas de desempenho da LSTM-AE para o primeiro teste.	40
Tabela 15 - Métricas de desempenho da LSTM-AE para o segundo teste.	40

LISTA DE ABREVIATURAS E SIGLAS

AE	<i>Autoencoder</i>
GRU	<i>Gated Recurrent Unit</i>
IA	Inteligência Artificial
IoT	<i>Internet of Things</i>
LSTM	<i>Long Short-Term Memory</i>
MAE	<i>Mean Absolute Error</i>
ML	<i>Machine Learning</i>
MLP	<i>Multilayer Perceptron</i>
ReLU	<i>Rectified Linear Unit</i>
RMSE	<i>Root Mean Squared Error</i>
RNA	Rede Neural Artificial
RNN	Redes Neurais Recorrentes

SUMÁRIO

1 INTRODUÇÃO.....	13
2 OBJETIVOS	15
2.1 Geral.....	15
2.2 Específicos	15
3 REVISÃO BIBLIOGRÁFICA	16
3.1 Indústria Açucareira e o Processo Fermentativo	16
3.2 <i>Soft-Sensors</i>	20
3.3 <i>Machine Learning</i>	21
3.4 Redes Neurais Artificiais.....	23
3.5 <i>Autoencoders</i>.....	26
4 METODOLOGIA.....	28
4.1 Simulação do Processo Fermentativo	28
4.2 Coleta de Dados	29
4.3 Pré-processamento dos Dados	29
4.4 Construção das Redes	31
4.5 Análise Comparativa.....	32
5 RESULTADOS E DISCUSSÃO.....	34
5.1 Simulação do Processo Fermentativo	34
5.2 MLP	36
5.3 LSTM.....	37
5.4 <i>LSTM-AUTOENCODER</i>	40
6 CONCLUSÃO.....	42
REFERÊNCIAS	43

1. INTRODUÇÃO

Fermentação alcoólica é um processo microbiano no qual bactérias ou leveduras convertem açúcares em etanol e outros compostos, sendo o etanol um produto com alto valor agregado e utilizado em bebidas alcoólicas, indústrias de perfumaria e principalmente como combustível (JÚNIOR, 2012). O Brasil é o segundo maior produtor de etanol do mundo, com 27,5% do volume global em 2021 (VIDAL, 2022).

A via fermentativa é a principal maneira de obtenção do etanol no Brasil, sendo a cana-de-açúcar a principal matéria-prima. O processo consiste resumidamente em fermentar o caldo e o mosto produzidos nas indústrias açucareiras e posteriormente levá-lo ao processo de destilação e desidratação, para o caso da produção do etanol anidro (JÚNIOR, 2012). O processo fermentativo depende de fatores físicos (temperatura e pressão osmótica), químicos (reações, nível de oxigenação, quantidade de nutrientes e inibições, pH) e microbiológicos (espécie e concentração do microrganismo, presença de contaminantes) (LIMA, 2019).

O controle de bioprocessos é uma tarefa complexa devido à quantidade de variáveis a serem controladas simultaneamente e também devido a dinâmica não linear dos sistemas. Processos convencionais de análise do produto e controle das variáveis que envolvem análises laboratoriais podem ser ineficientes devido a elevados tempos de resposta das análises e imprecisões associadas aos instrumentos e erros humanos (DESAI et al., 2006). Uma alternativa em relação a medidas convencionais é a utilização de *soft-sensors*, que se baseiam em dados do processo, normalmente variáveis secundárias, para prever ou inferir o comportamento de variáveis mais críticas, também chamadas de variáveis primárias (WANG et al., 2019).

Soft-sensors frequentemente usam redes neurais, incluindo *Multilayer Perceptron* (MLP) e *Long Short-Term Memory* (LSTM). As LSTM são favoráveis para processos dinâmicos porque capturam dependências temporais, enquanto as MLP modelam correlações não lineares entre variáveis. Para processos com um grande número de variáveis esse tipo de sensor pode ser utilizado paralelamente com *autoencoder* (AE) para redução de dimensionalidade, compressão de dados e também detecção de *outliers*, podendo melhorar a eficiência dos sensores (KAY et al., 2022).

A indústria 4.0 representa a evolução do setor industrial impulsionada pela implementação de tecnologias como a Internet das Coisas (IoT, em inglês), sistemas físico-cibernéticos e inteligência artificial (IA). Estas tecnologias permitem que os sistemas aprendam a partir da experiência utilizando dados do processo e se adaptem a novas situações (BANITAAN et al.,

2023). Empresas como a Dow e Shell tem casos de sucesso utilizando IA em seus processos. A Dow desenvolveu uma capacidade de Inteligência Preditiva que transforma o desenvolvimento de produtos de poliuretanos, acelerando em até 200 mil vezes a formulação de novos produtos e reduzindo o tempo de descoberta para cerca de 30 segundos. Já a Shell aplicou IA na análise de dados das bombas submersas de sua plataforma Perdido no Golfo do México, identificando um padrão que antecipa falhas em 70% dos casos, prevenindo interrupções de produção e aumentando a eficiência operacional.

Um tema de interesse tanto para a comunidade acadêmica quanto para a indústria é a utilização de *soft-sensors* em conjunto com sistemas inteligentes ou IA, como os *deep learning autoencoders*, para a estimativa de variáveis em processos fermentativos. Dessa forma, o objetivo deste trabalho é simular o bioprocesso de fermentação alcoólica em batelada alimentada a fim de gerar dados emulados do processo e aplicar técnicas de IA, como MLP e *deep learning autoencoders*, para desenvolver um *soft-sensor*. Além disso, será avaliada a eficácia de estratégias para a elaboração do *soft-sensor* em processos em batelada que apresentam particularidades na representação da variável tempo, utilizando as redes LSTM.

2. OBJETIVOS

2.1. Geral

Desenvolver um *soft-sensor*, em linguagem Python, eficaz para predição de variáveis do processo de fermentação alcoólica em batelada alimentada.

2.2. Específicos

- Simular o processo de fermentação alcoólica em batelada alimentada, em linguagem Python, utilizando um modelo já validado com dados experimentais;
- Desenvolver um *soft-sensor* que realize inferências sobre variáveis de interesse do processo utilizando as redes MLP e LSTM;
- Utilizar a *deep learning autoencoder* para melhorar as relações entre os dados gerados da simulação e avaliar sua influência na acuracidade do sensor.

3. REVISÃO BIBLIOGRÁFICA

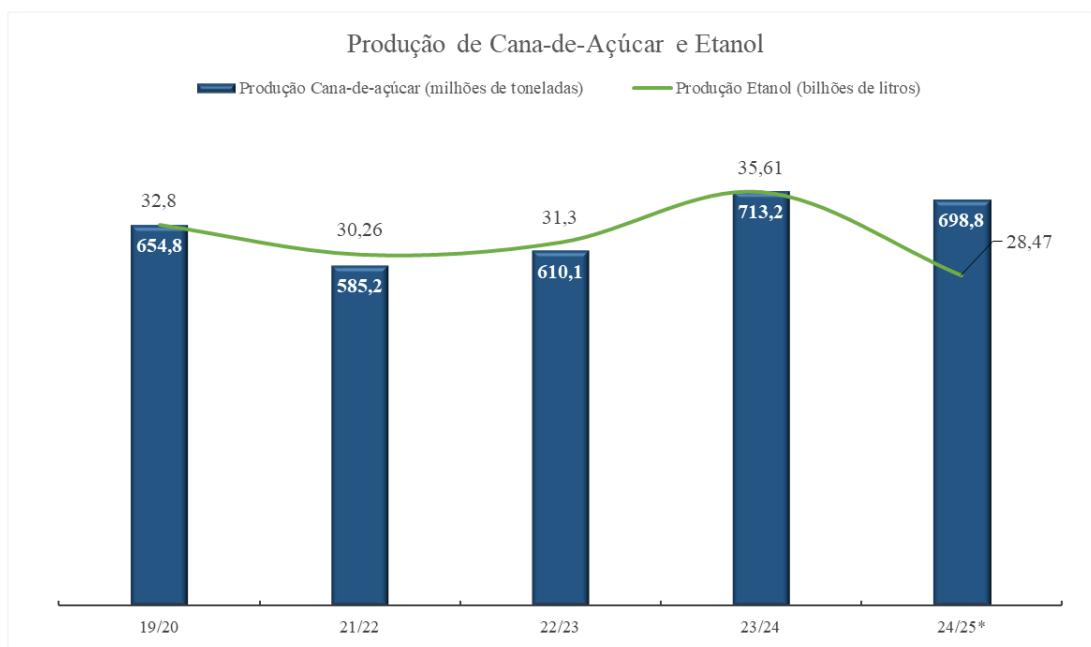
Na revisão será feita uma análise sobre o atual mercado açucareiro e de álcool no Brasil e sobre o processo fermentativo em batelada alimentada. Serão abordados tópicos sobre *soft-sensor* e as suas aplicações, *machine learning*, redes neurais artificiais, *autoencoders* e as suas utilizações em processos.

3.1. Indústria Açucareira e o Processo Fermentativo

O etanol pode ser obtido por três vias básicas, como a destilatória, a sintética e por bioprocesso, sendo a principal e mais utilizada no Brasil a última. O processo fermentativo utiliza como matérias-primas vegetais ricos em carboidratos, que podem ser classificados como sacarinos, amiláceos e celulósicos (LIMA, 2019).

O Brasil é o maior produtor de açúcar e o segundo maior produtor de etanol do mundo, e isso é consequência de o país ter condições climáticas ótimas para cultivo da cana-de-açúcar e investimentos no setor sucroalcooleiro. O país registra uma produção de 713,2 milhões de toneladas de cana-de-açúcar na safra 2023/2024, com a produção de etanol superando 35 bilhões de litros, tendo um aumento de 15% em relação ao ano anterior (CONAB, 2024; VIDAL, 2022).

Figura 1 - Produção de cana-de-açúcar e etanol entre o período de 2019 e 2024. A informação para a safra 24/25 é uma estimativa.



Fonte: Autor (2024).

A obtenção do álcool por via fermentativa pode ser dividida em três fases: preparo do substrato, a fermentação e a destilação. Esse processo pode acontecer em diferentes regimes como em batelada, batelada alimentada, semicontínuo, contínuo e em estado sólido, sendo o foco deste trabalho a fermentação em batelada alimentada. No processo descontínuo alimentado, um ou mais nutrientes são adicionados ao fermentador durante o cultivo e os produtos permanecem até o final da operação. A vazão de alimentação de nutrientes pode variar com o tempo e a do mosto pode se dar de forma contínua ou intermitente e essa exerce grande influência sobre o rendimento do processo (SCHMIDELL, 2001; COELHO, 2009). Como já mencionado, o processo fermentativo depende de fatores físicos, químicos e microbiológicos, tais como: temperatura, pressão, nível de oxigenação, presença de inibidores e contaminantes. As principais condições que influenciam na cinética da população celular são as reações em solução, pH, temperatura, reologia do meio, quantidade de nutrientes e uniformidade no reator (BAILEY, 1986).

A modelagem matemática do processo fermentativo em batelada alimentada se baseia no que foi proposto por Vasconcelos (1987) e validado por Coelho (2009) com dados experimentais em escala piloto e industrial. As seguintes considerações são feitas:

- Sistema com mistura perfeita com densidade constante;
- Misturas do mosto e do meio de fermentação consideradas diluídas;
- Sistema isotérmico;
- Consumo de substrato para manutenção celular desprezível;
- Sem perdas de etanol por arraste com CO₂ ou evaporação.

Considerando isso, o balanço de massa global do processo é dado pela Equação 1, onde a variação de massa do meio depende apenas da alimentação.

$$\frac{d(\rho V)}{dt} = \rho_a F \xrightarrow{\rho=\rho_a} \frac{dV}{dt} = F \quad (1)$$

Para as células (Equação 2) considera-se que não há morte celular nem alimentação durante todo o processo, então a variação depende apenas da taxa de crescimento celular r_X e do volume, sendo assim:

$$\frac{d(XV)}{dt} = r_X V \quad (2)$$

Para as concentrações de substrato e etanol, os balanços são mostrados nas Equações 3 e 4, onde S , S_a e E são as concentrações de substrato no meio, substrato na alimentação e etanol, respectivamente.

$$\frac{d(SV)}{dt} = S_a F - r_s V \quad (3)$$

$$\frac{d(EV)}{dt} = r_E V \quad (4)$$

Manipulando as equações 2, 3 e 4 a fim de isolarmos as variáveis de interesse temos:

$$\frac{dX}{dt} = r_X - \frac{X}{V} \frac{dV}{dt} \quad (5)$$

$$\frac{dS}{dt} = S_a \frac{F}{V} - r_s - \frac{S}{V} \frac{dV}{dt} \quad (6)$$

$$\frac{dE}{dt} = r_E - \frac{E}{V} \frac{dV}{dt} \quad (7)$$

Utilizando a Equação 1 e o fator de diluição, definido como D , onde $D = F/V$, obtém-se:

$$\frac{dX}{dt} = r_X - XD \quad (8)$$

$$\frac{dS}{dt} = D(S_a - S) - r_s \quad (9)$$

$$\frac{dE}{dt} = r_E - ED \quad (10)$$

Usando as expressões para as taxas, nas quais $r_X = \mu X$, $r_s = \sigma X$ e $r_E = \gamma X$, em que μ , σ e γ são as velocidades específicas de crescimento celular, consumo de substrato e formação de etanol, respectivamente, o balanço se resume às Equações 11, 12, 13 e 14.

$$\frac{dV}{dt} = F \quad (11)$$

$$\frac{dX}{dt} = (\mu - D)X \quad (12)$$

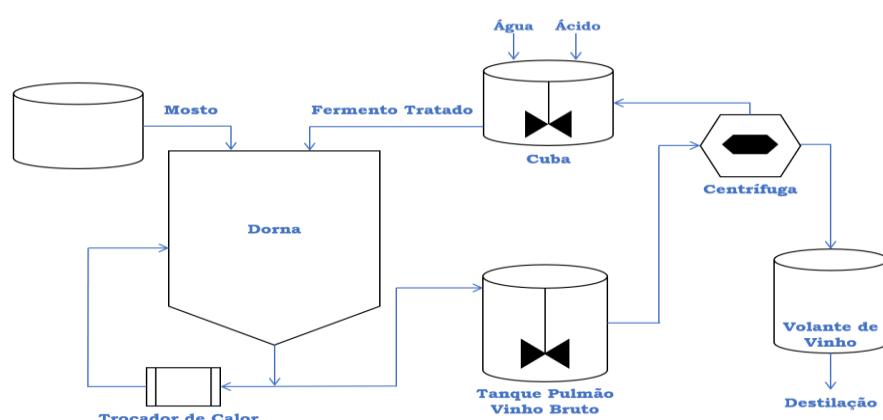
$$\frac{dS}{dt} = D(S_a - S) - X \left(\frac{\mu}{Y_x \frac{s}{s}} + \frac{\gamma}{Y_p \frac{s}{s}} \right) \quad (13)$$

$$\frac{dE}{dt} = \gamma X - ED \quad (14)$$

Em resumo, as concentrações X , S e E , dadas em gramas por litro (g/L), são, respectivamente, a concentração de células (leveduras), a concentração de substrato e a concentração de etanol. A variável V é o volume da dorna e é dada em litros (L). O parâmetro D representa o efeito de diluição provocado pelo aumento de volume na dorna e sua unidade é h^{-1} . F é o perfil de alimentação do mosto, dado em litros por segundo (L/s). Os parâmetros adimensionais $Y_x \frac{s}{s}$ e $Y_p \frac{s}{s}$ representam os fatores de rendimento substrato-célula e substrato-etanol, respectivamente.

Os parâmetros μ e γ (h^{-1}) representam as velocidades específicas de crescimento e dependem do modelo cinético (Coelho, 2009).

Figura 2 - Representação de um processo fermentativo em batelada alimentada.



Fonte: Autor (2024).

Alguns dos diversos modelos cinéticos que buscam descrever o comportamento da fermentação existentes na literatura estão dispostos na Tabela 1. Eles buscam descrever o comportamento da fermentação considerando ou não a inibição por substrato ou produto. O modelo de Monod, diferente do modelo de Teissier, não leva em consideração a inibição por substrato,

na medida em que os modelos de Hinshelwood, Aiba e Vasconcelos levam em consideração a inibição por produto. As constantes K_S , $K_{S'}$, K_P e $K_{P'}$ são constantes cinéticas obtidas a partir de experimentos.

Tabela 1 - Modelos cinéticos.

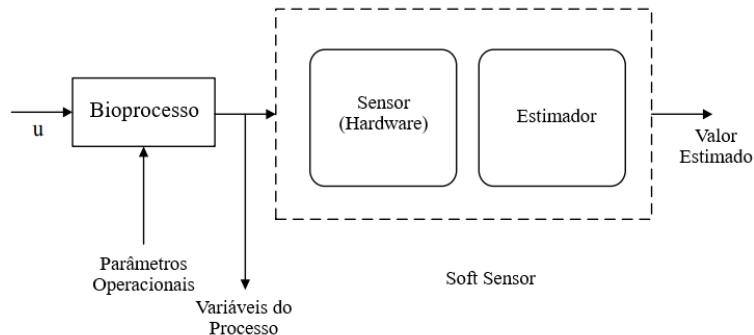
Modelo	μ	γ
Monod	$\mu_{\max} \cdot \left(\frac{S}{K_S + S} \right)$	$\gamma_{\max} \cdot \left(\frac{S}{K_{S'} + S} \right)$
Teissier	$\mu_{\max} \cdot \left(1 - \exp \left(-\frac{S}{K_S} \right) \right)$	$\gamma_{\max} \cdot \left(1 - \exp \left(-\frac{S}{K_{S'}} \right) \right)$
Hinshelwood	$\mu_{\max} \cdot \left(\frac{S}{K_S + S} \right) \cdot (1 - K_P \cdot E)$	$\gamma_{\max} \cdot \left(\frac{S}{K_{S'} + S} \right) \cdot (1 - K_{P'} \cdot E)$
Aiba, Shoda e Nagatani	$\mu_{\max} \cdot \left(\frac{S}{K_S + S} \right) \cdot \exp(K_P \cdot E)$	$\gamma_{\max} \cdot \left(\frac{S}{K_{S'} + S} \right) \cdot \exp(K_{P'} \cdot E)$
Vasconcelos	$\mu_{\max} \cdot \left(\frac{S}{K_S + S} \right) \cdot \left(1 - \frac{E}{K_P} \right)$	$\gamma_{\max} \cdot \left(\frac{S}{K_{S'} + S} \right) \cdot \left(1 - \frac{E}{K_{P'}} \right)$

Fonte: Adaptado de Belo (2021).

3.2. *Soft-Sensors*

Um *soft-sensor* é um sensor que utiliza inferência estatística para determinar o valor de variáveis primárias de difícil medição a partir de variáveis secundárias fáceis de mensurar. Na maioria das vezes esse tipo de sensor utiliza modelos empíricos para correlacionar as variáveis dos processos e são especialmente utilizados em processos complexos, nos quais não existem modelos matemáticos e/ou fenomenológicos que definam o sistema. Sua principal função é substituir sensores convencionais, que podem ter um custo de aquisição e manutenção elevado, e análises laboratoriais de amostras coletadas do processo que, a depender da natureza da análise, podem ter um alto tempo de resposta e a chance de erros humanos (YAN, W., SHAO, G., WANG, X., 2004; LOTUFO, F. A., GARCIA, C., 2008). O conceito de um *soft-sensor* pode ser visto na Figura abaixo (Figura 3).

Figura 3 - Modelo adaptado de um *soft-sensor* com um *hardware* e um estimador.



Fonte: Adaptado de Zhu (2020).

A Tabela 2 mostra diversas aplicações de *soft-sensor* em diversos setores industriais.

Tabela 2 - Exemplos de aplicação de *soft-sensors* em processos químicos.

Publicação	Indústria	Aplicação
Zhao et al. (2021)	Indústria de Cimento	Previsão do conteúdo de óxido de cálcio livre em um clínquer de cimento
Farahani et al. (2021)	Usina Elétrica	Previsão da potência ativa e fluxo de combustível
Wang et al. (2019b)	Indústria Siderúrgica	Previsão da operação de têmpera a seco de coque
Yan et al. (2020)	Estação de Tratamento de Águas Residuais	Previsão do nitrogênio Kjeldahl total
Phatwong and Koolpiruck (2019)	Indústria de Papel e Celulose	Previsão do número Kappa de um digestor de celulose
Sun and Ge (2019)	Processo de Síntese de Amônia	Previsão da concentração de CO ₂ em uma coluna de absorção de CO ₂
Liu et al. (2021a)	Indústria de Processamento de Polímeros	Previsão do índice de fluxo de fusão (MFI) em um processo de polimerização de polipropileno
Guo et al. (2020b)	Indústria Petroquímica	Previsão do conteúdo de butano em uma coluna de debutanizador de uma refinaria de petróleo
Qiu et al. (2021)	Indústria Farmacêutica	Previsão da concentração de penicilina em um processo de fermentação de penicilina
Meng et al. (2019)	Indústria de Processamento de alimentos	Previsão da pureza do líquido-mãe e supersaturação em um processo de cristalização de açúcar de cana

Fonte: Adaptado de Perera, Y.S. et al (2023).

A indústria 4.0 é impulsionada pela implementação deste tipo de tecnologia para o controle e otimização de processos em tempo real. Ela permite a monitorização de equipamentos industriais através de sensores, combinando dados históricos com análises preditivas para otimizar a manutenção e prevenir falhas.

3.3. Machine Learning

Machine Learning (ML) é um campo da ciência da computação que foca na aplicação de algoritmos para resolver problemas da vida real. Esses algoritmos se baseiam em informações do problema para criar modelos estatísticos capaz de realizar previsões. Existem diferentes

tipos de algoritmos que definem como o aprendizado acontece, sendo de forma supervisionada, semi-supervisionada, não supervisionada e de reforço.

No aprendizado supervisionado o algoritmo aprende a partir de um conjunto de dados definidos, onde as entradas tem correlações com as saídas, e então o modelo tenta prever as saídas corretamente. As principais aplicações são para regressão, que é o caso deste trabalho, e classificação. O aprendizado não supervisionado não rotula os dados e, portanto, não há resultados pré-definidos para o modelo utilizar como referência para aprender. Normalmente estes modelos são utilizados para descobrir padrões entre os dados e suas principais aplicações são para *clustering* e redução de dimensionalidade, como é o caso do AE, foco deste estudo. Já no aprendizado por reforço o modelo é treinado para aprender uma política de ação que recebe recompensas ou penalidades com base nas ações realizadas, se adaptando para maximizar a função de recompensa. Exemplos de aplicação dos aprendizados estão na Figura 4.

Figura 4 - Exemplos de aplicação em função do tipo de aprendizado.



Fonte: DIO. Diferenças entre aprendizado supervisionado e não supervisionado¹.

Um *soft-sensor* é uma aplicação específica de aprendizado de máquina em sistemas industriais para prever as variáveis do processo utilizando os dados disponíveis (BURKOV, 2019).

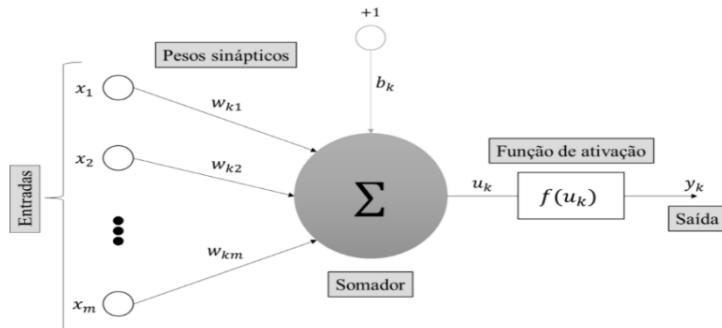
¹ Disponível em: <https://www.dio.me/articles/diferencias-entre-aprendizado-supervisionado-e-nao-supervisionado>. Acesso em: 10 nov. 2024.

3.4. Redes Neurais Artificiais

Redes Neurais Artificiais (RNA) são modelos matemáticos que são aplicados como ML e que são baseadas em como o neurônio e cérebro humano funcionam (BELO, 2021). Devido ao avanço da capacidade de processamento de dados dos computadores modernos, a aplicação de RNAs ganhou espaço em muitas áreas como na economia, saúde, finanças e processos indústrias, sendo aplicados a *soft-sensor*. As principais vantagens do uso de RNAs são a velocidade e volume de processamento, tolerância a erros, escalabilidade e adaptabilidade (ABIODUN et al., 2018).

A unidade principal de processamento de uma rede neural é o neurônio e ele tem três elementos básicos, sendo: um conjunto de sinapses, que são caracterizados por pesos; um somador, que soma os sinais de entrada e uma função de ativação que limita a amplitude de saída do neurônio (HAYKIN, 1999). A Figura a seguir (Figura 5) mostra o esquema de um modelo de neurônio.

Figura 5 - Neurônio artificial e seus elementos básicos.



Fonte: Belo (2021).

A saída do neurônio (Equação 15) é definida como uma função aplicada a soma ponderada do produto escalar do vetor de entradas (x) pelo vetor de pesos (w), acrescido do *bias* (b), que tem como função ajustar a saída do neurônio. A função que transforma a soma é chamada de função de ativação e tem como objetivo introduzir a não linearidade aos cálculos (HAYKIN, 1999; CASTRO, 2007).

$$y_k = f \left(\sum_{j=1}^m w_{kj} x_j + b_k \right) = f(w \cdot x + b) \quad (15)$$

Existem diversos tipos de funções de ativação e cada uma tem uma aplicação específica, conforme exibido na tabela abaixo (Tabela 3).

Tabela 3 - Funções de ativação e suas aplicações.

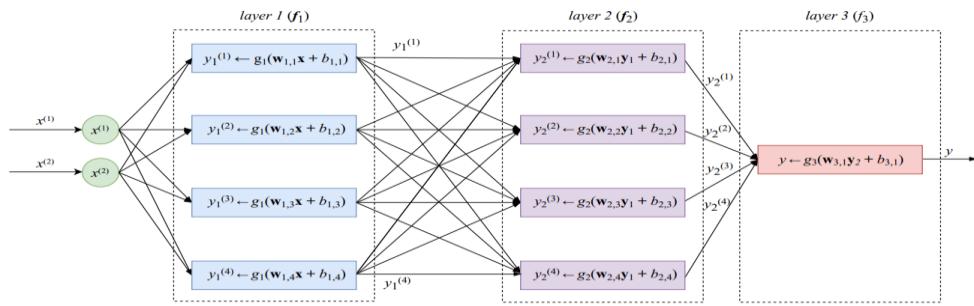
Função	Equação	Aplicação
Linear	$f(u_k) = u_k$	Utilizada em regressão, principalmente nas camadas de saída
Degrau	$f(u_k) = \begin{cases} 1 & \text{se } u_k \geq 0 \\ 0 & \text{se } u_k < 0 \end{cases}$	Utilizada para classificação de padrões com separação linear
Sigmoid	$f(u_k) = \frac{1}{1 + e^{-u_k}}$	Como a degrau, é adequada para classificação binária, porém introduz não linearidade
Tangente Hiperbólica (tanh)	$f(u_k) = \frac{e^{u_k} - e^{-u_k}}{e^{u_k} + e^{-u_k}}$	Utilizada quando valores negativos tem significância
<i>Rectified Linear Unit</i> (ReLU)	$f(u_k) = \max(0, u_k)$	Muito eficiente e utilizada para redes profundas pois evita o problema de <i>Vanishing Gradient</i>

Fonte: Autor (2024).

Uma arquitetura específica de rede neural é a *multilayer perceptron* (Figura 6). A rede é a combinação de unidades lógicas organizadas em uma ou mais camadas que recebem as informações de entrada, aplicam uma operação matemática e geram uma saída. O treinamento desse tipo de rede utiliza o algoritmo chamado de retropropagação de erros (*backpropagation*, em inglês), geralmente combinado com o método de otimização chamado de gradiente descendente. Inicialmente a saída da rede é calculada com base nas entradas e em seguida o gradiente da função objetivo, como, por exemplo, o erro quadrático, é computado. Os pesos são atualizados subtraindo-se o gradiente, ajustado pela taxa de aprendizado, para corrigir as conexões entre a camada de saída e a última camada oculta. Esse processo de ajuste prossegue de forma iterativa, retrocedendo até a camada de entrada. O ciclo completo de cálculo da saída, ajuste dos pesos e reinício do processo é chamado de época (*epoch*, em inglês). Durante o treinamento, os erros das etapas de treino e teste são registrados e frequentemente apresentados em gráficos para monitorar o desempenho e evitar sobreajustes (SOARES, 2017).

Na Figura 6 é possível ver uma MLP de 3 camadas, com duas camadas ocultas e uma de saída. Essa rede pode ser um modelo de classificação ou regressão, a depender da função de ativação usada na terceira camada.

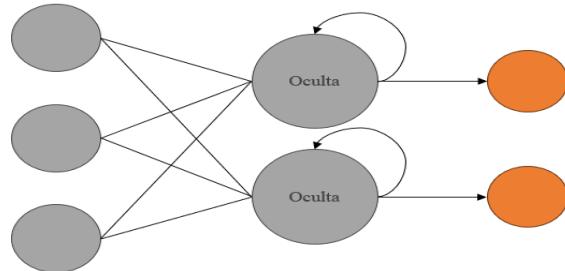
Figura 6 - Representação de uma rede neural MLP de 3 camadas.



Fonte: Burkov (2019).

Um dos tipos de redes neurais são as redes neurais recorrentes (RNNs) e são utilizadas para classificar e gerar sequências, como séries temporais, e devido a isso suas principais aplicações são em processamento de texto e fala e tratamento de dados temporais, que é o interesse deste trabalho. Esse tipo de rede contém *loops* e cada camada contém um valor real de estado (memória da unidade). Cada unidade em cada camada recebe duas entradas que são: um vetor de estados da camada anterior e o estado da camada atual (Figura 7) (BURKOV, 2019). Um problema que redes desse tipo têm é a dificuldade de lidar com entradas muito longas porque as entradas dos vetores do início tendem a ser esquecidas devido ao estado de cada unidade.

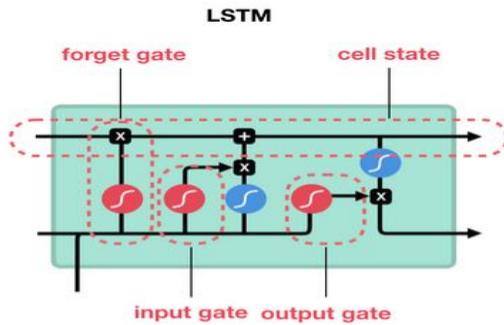
Figura 7 - Rede Neural Recorrente Simples.



Fonte: Autor (2024).

Para contornar esse problema foram desenvolvidos modelos de máquinas de aprendizado específicas para dados temporais como o *gated recurrent neural network* (GRU) e *long short-term memory* (BURKOV, 2019). A arquitetura da rede LSTM (Figura 8) a ser utilizada nesse trabalho consiste em sub redes neurais conectadas, conhecidas como blocos de memória, que mantêm o estado de cada camada e regula a informação que é passada (HOUDET, G. V., MOSQUERA, C., NÁPOLES, G., 2020).

Figura 8 - Arquitetura típica de um bloco LSTM. A rede é composta por um portão de entrada (*input gate*), um portão de esquecimento (*forget gate*), a célula (*cell*) e o portão de saída (*output gate*).



Fonte: Deep Learning Book².

O portão de entrada de uma rede LSTM determina as informações que devem ser armazenadas na célula de memória. O portão de esquecimento decide quais informações devem ser mantidas ou esquecidas e o portão de saída filtra as informações com base no estado da célula, controlando a saída para a próxima etapa. Conforme destaca Houdt (2020), esse tipo de rede tem a capacidade de lidar com diversos cenários de previsão de séries temporais e pode ser aplicado em diversas áreas como previsões do mercado financeiro, produção de petróleo, preço do petróleo e diagnósticos de falhas.

3.5. Autoencoders

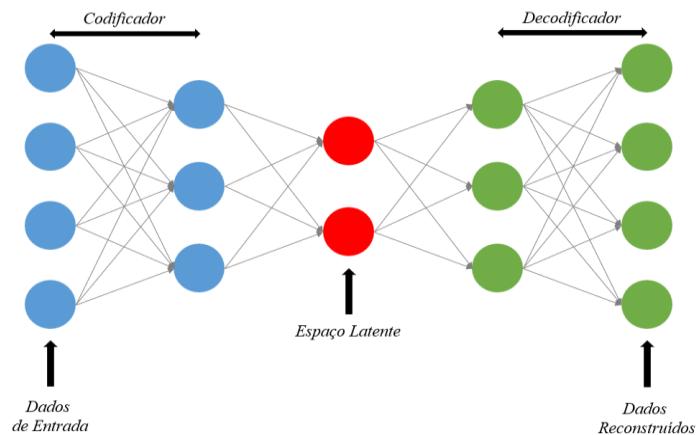
Outro tipo de RNA é *autoencoder*, que é uma rede de aprendizado não supervisionado na qual sua principal função é reconstruir os dados de entrada, extraíndo as informações mais relevantes (LIMA, 2021). Sua estrutura básica (Figura 9) consiste em um codificador, que compacta os dados de entrada para o espaço latente de menor dimensionalidade, e um decodificador, que reconstrói a entrada a partir do código latente. A reconstrução é otimizada para que o valor reconstruído seja o mais próximo possível do original. A camada compactada atua como uma representação comprimida das características mais importantes do dado original (BURKOV, 2019). A otimização do processo de reconstrução dos dados é feita reduzindo o erro quadrático médio entre o dado previsto e o real e é dado por:

$$\frac{1}{N} \left(\sum_{i=1}^N \|x_i - f(x_i)\|^2 \right) \quad (16)$$

² Disponível em: <https://www.deeplearningbook.com.br/arquitetura-de-redes-neurais-gated-recurrent-unit-gru/>. Acesso em: 17 de jan. de 2024.

Esse tipo de rede é muito usado para redução de dimensionalidade, remoção de ruídos em dados, geração de dados sintéticos e problemas de detecção de anomalias (KIM, S. G., CHAE, Y. G., SEONG, P. H., 2019). A utilização de *autoencoders* como ferramenta no desenvolvimento de *soft-sensor* vem sendo estudada ultimamente, como pode ser destacado nos trabalhos de Kay (2022), que utiliza a rede para redução de dimensionalidade e melhora de performance na predição de viscosidade do produto, e no trabalho de Menegolla (2019), que utiliza para inferências em processos químicos. Neste trabalho o AE será utilizado para redução de dimensionalidade e será avaliado sua influência na capacidade do modelo em realizar as inferências das variáveis do processo fermentativo em batelada alimentada.

Figura 9 - Estrutura básica de um *autoencoder*.

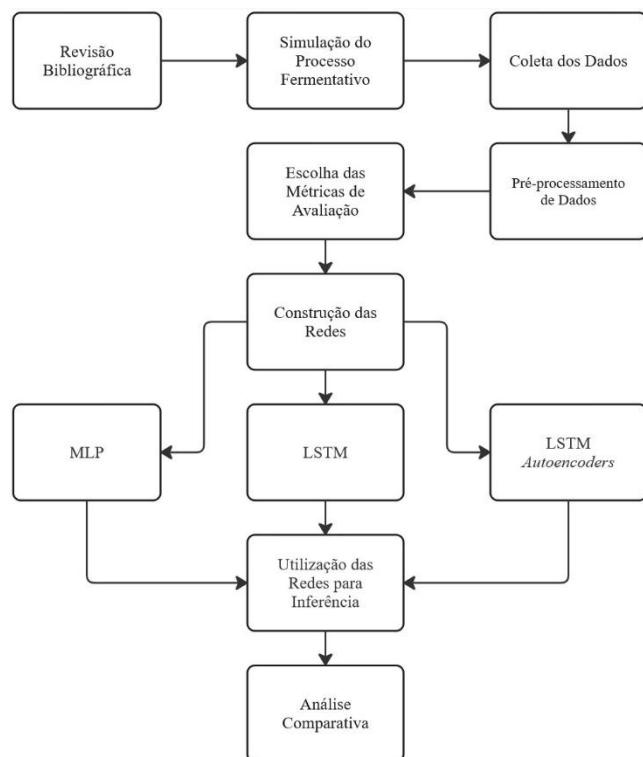


Fonte: Autor (2024).

4. METODOLOGIA

A metodologia para realização desse trabalho pode ser resumida conforme Figura 10. Primeiramente ocorreu a revisão bibliográfica sobre a construção de modelos de redes neurais em linguagem Python, com foco nas MLP, LSTM e AE e também sobre processo fermentativo. Em seguida, foi utilizado o método Runge Kutta de quarta ordem para resolução das equações diferenciais resultantes do balanço de massa do processo de fermentação alcoólica em batelada alimentada.

Figura 10 - Fluxo da metodologia do trabalho.



Fonte: Autor (2024).

4.1. Simulação do Processo Fermentativo

Para simulação do processo foram utilizadas as bibliotecas SciPy e NumPy para resolução das equações diferenciais e Matplotlib para elaboração dos gráficos. As variáveis que tiveram seus valores alterados para realização das simulações foram a vazão de alimentação e concentração inicial do substrato e seus valores estão na tabela abaixo (Tabela 4). O modelo cinético utilizado foi o modelo de Teissier. No total foram feitas dez simulações, sendo nove

para geração de dados que foram utilizados para treinar o modelo e uma para testá-lo. A Tabela 5 mostra os valores das outras variáveis do processo.

Tabela 4 - Valores das variáveis em cada simulação.

Simulação	Vazão de Alimentação (L/s)	Concentração Inicial do Mosto (g/L)
1	7	88,0
2	7	132,5
3	7	177,0
4	13	88,0
5	13	132,5
6	13	177,0
7	19	88,0
8	19	132,5
9	19	177,0
10 (teste)	12	150,0

Fonte: Autor (2024).

Tabela 5 - Valores das variáveis do processo.

Variável	Valor Utilizado	Unidade
K_{sx}	168,2761	g/L
K_{sp}	43,5830	g/L
$Y_{x/s}$	0,51	-
$Y_{p/s}$	0,57	-
$u_{MÁX}$	0,0022356	s ⁻¹
$\gamma_{MÁX}$	0,0008194	s ⁻¹
V_0	4375	L
X_0	30	g/L

Fonte: Autor (2024).

4.2. Coleta de Dados

O tempo utilizado para cada simulação de batelada foi de dez horas, com a alimentação de mosto acontecendo somente nas seis primeiras horas. Foi feita a avaliação dos modelos usando um minuto como tempo de amostragem, gerando assim 600 linhas de dados por batelada contendo 12 variáveis, sendo as variáveis tempo, vazão, concentração inicial de mosto, e as concentrações de etanol, células e substrato com atraso (*lag*, em inglês).

4.3. Pré-processamento dos Dados

A Tabela 6 mostra os valores de uma variável arbitrária A ao longo do tempo. Imaginando que queremos usar os três valores anteriores ($lag = 3$) da variável para prever o próximo, devemos organizar os dados da seguinte forma, conforme ilustrado na Tabela 7. A tabela mostra a sequência de informações que deve ser fornecida ao modelo para que ele preveja o próximo valor. A última linha da tabela mostra o valor esperado da sequência com base no histórico.

Tabela 6 - Valores de uma variável arbitrária A ao longo do tempo.

Tempo (min)	Valor da variável A
0	0
5	1
10	2
15	3
20	4
25	5
30	6
35	7
40	8
45	9
50	10

Fonte: Autor (2024).

Tabela 7 - Exemplo de como os dados são organizados e inseridos na rede.

Sequência de Entrada (Lag 3)	Target (Próximo valor)
[0, 1, 2]	3
[1, 2, 3]	4
[2, 3, 4]	5
[3, 4, 5]	6
[4, 5, 6]	7
[5, 6, 7]	8
[6, 7, 8]	9
[7, 8, 9]	10
[8, 9, 10]	11 (valor esperado)

Fonte: Autor (2024).

Essa mesma metodologia foi aplicada nos dados do processo que foram utilizados na MLP e LSTM. Como queremos prever o valor das variáveis de interesse, foi aplicado um lag de 3 nelas e os dados foram divididos em dois arquivos, um contendo os dados de entrada e no outro os dados do próximo valor, conforme exibido na Tabela 8. Foi avaliado também a eficácia do modelo usando somente a variável de interesse (etanol) nos dados de entrada e de saída.

Tabela 8 – Variáveis alimentadas em cada teste. Os subscritos 1, 2 e 3 representam o atraso da variável.

Teste	Dados de Entrada	Dados de Saída (targets)
1	$t, F, S_a, E_1, E_2, E_3, S_1, S_2, S_3, X_1, X_2, X_3$	E, S, X
2	t, F, S_a, E_1, E_2, E_3	E

Fonte: Autor (2024).

Uma das etapas mais importantes em qualquer atividade relacionada à ciência de dados é o pré-processamento, que inclui normalização, tratamento de valores ausentes, redução de dimensionalidade, remoção de *outliers*, entre outros (ALIABADI, 2020). Como os dados foram obtidos da simulação do processo, foi somente preciso normaliza-los. Para normalização foi utilizado o método **MinMaxScaler** da biblioteca *Scikit-learn*, que reescalou os dados para um intervalo específico, normalmente entre 0 e 1, conforme mostra a Equação 17. Essa metodologia preserva a relação entre os valores originais e facilitando o desempenho do modelo. É uma abordagem vantajosa especialmente em redes neurais, onde a escala dos dados pode impactar o processo de aprendizado (BURKOV, 2019).

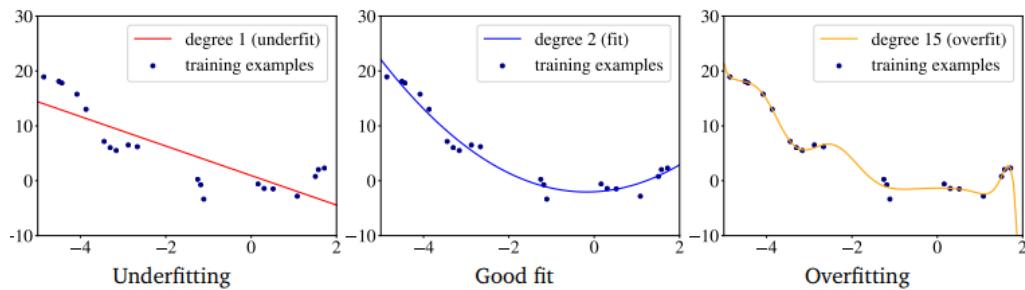
$$X_{\text{normalizado}} = \frac{X_{\text{MÁX}} - X_{\text{MÍN}}}{X - X_{\text{MÍN}}} \quad (17)$$

Onde X é o valor original, $X_{\text{MÍN}}$ é o valor mínimo do conjunto de dados, $X_{\text{MÁX}}$ é o valor máximo e X o valor original.

4.4. Construção das Redes

Para construção de todas as redes foram utilizadas as bibliotecas *Scikit-learn*, *TensorFlow* e *Keras*. Foram utilizadas técnicas para evitar o fenômeno de *overfitting* (sobreajuste), exemplificado na Figura 11, no qual o modelo se adapta excessivamente aos dados de treinamento e posteriormente não consegue realizar previsões precisas nos dados de teste (BURKOV, 2019).

Figura 11 - Exemplos visuais de ajustes aos dados. A figura mais à esquerda representa uma falta de ajuste enquanto que a mais à direita representa o sobreajuste. A figura central representa um bom ajuste.



Fonte: Burkov (2019).

As técnicas utilizadas foram a de *Dropout* e *Earlystopping*. A primeira é uma técnica de regularização que previne o *overfit* desativando aleatoriamente uma fração dos neurônios durante a fase de treinamento em cada iteração, fazendo com que estes neurônios não participem do cálculo e nem da retropropagação, melhorando assim a generalização do modelo. Neste trabalho foi utilizado um *dropout* (esquecimento) de 30% na MLP e 20% nas LSTM e LSTM-AE. A segunda técnica faz com que o treinamento do modelo seja interrompido caso o modelo não esteja mais melhorando seu desempenho no conjunto de validação. Neste caso a função de perda é monitorada em cada época (ciclo de treinamento e ajuste de parâmetros) e caso ela esteja variando pouco o treinamento é interrompido após um número definido de épocas, chamado de *patience* (paciência). Em todas as redes foi utilizado um valor de *patience* de 10. Para o tipo de treinamento foi utilizado o método de *mini-batch* em todos os modelos. Nesse método os dados são divididos e processados em *batchs* e o modelo atualiza os pesos após cada processamento. Para as funções de ativação foram utilizados a ReLU e a Linear, sendo a segunda apenas para a última camada. Todos os modelos utilizaram 2 camadas ocultas, sendo a primeira com 24 neurônios e a segunda com 12. As camadas iniciais e finais tiveram o mesmo tamanho das quantidades de variáveis dos vetores de entrada e saída, respectivamente.

A Tabela 9 mostra as arquiteturas das redes MLP, LSTM e LSTM-AE.

Tabela 9 – Parâmetros das redes.

Parâmetro	MLP	LSTM	LSTM-AE
Epochs (max.)	200	100	100
Dropout	30%	20%	20%
Patience	10	10	10
Taxa de Aprendizado	0.001	0.001	0.001
Tipo de Treinamento	<i>Mini-batch</i> (32)	<i>Mini-batch</i> (32)	<i>Mini-batch</i> (32)
Função de Ativação	ReLU e Linear	ReLU e Linear	ReLU e Linear

Fonte: Autor (2024).

4.5. Análise Comparativa

As métricas utilizadas para avaliar os modelos foram *mean absolute error* (MAE), *root mean squared error* (RMSE) e R^2 e suas formas de cálculo podem ser vistas nas Equações 18 e 19.

O MAE calcula a média dos erros absolutos entre as previsões e os valores reais, sendo uma métrica simples que mostra a magnitude média dos erros cometidos pelo modelo, sem considerar a direção dos erros (positivos ou negativos).

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

O RMSE mede a diferença entre os valores previstos pelo modelo e os valores reais e é obtido fazendo-se a raiz quadrada da média dos erros quadráticos. O RMSE penaliza erros maiores mais severamente devido à operação de quadrado.

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

Tanto para o MAE quanto para o RMSE, n representa o número de amostras, y_i o valor real, \hat{y}_i o valor predito.

Já o R^2 é uma métrica que indica a proporção da variância nos dados que é previsível a partir do modelo. Ele varia de 0 a 1, onde 1 indica que o modelo explica completamente a variância dos dados.

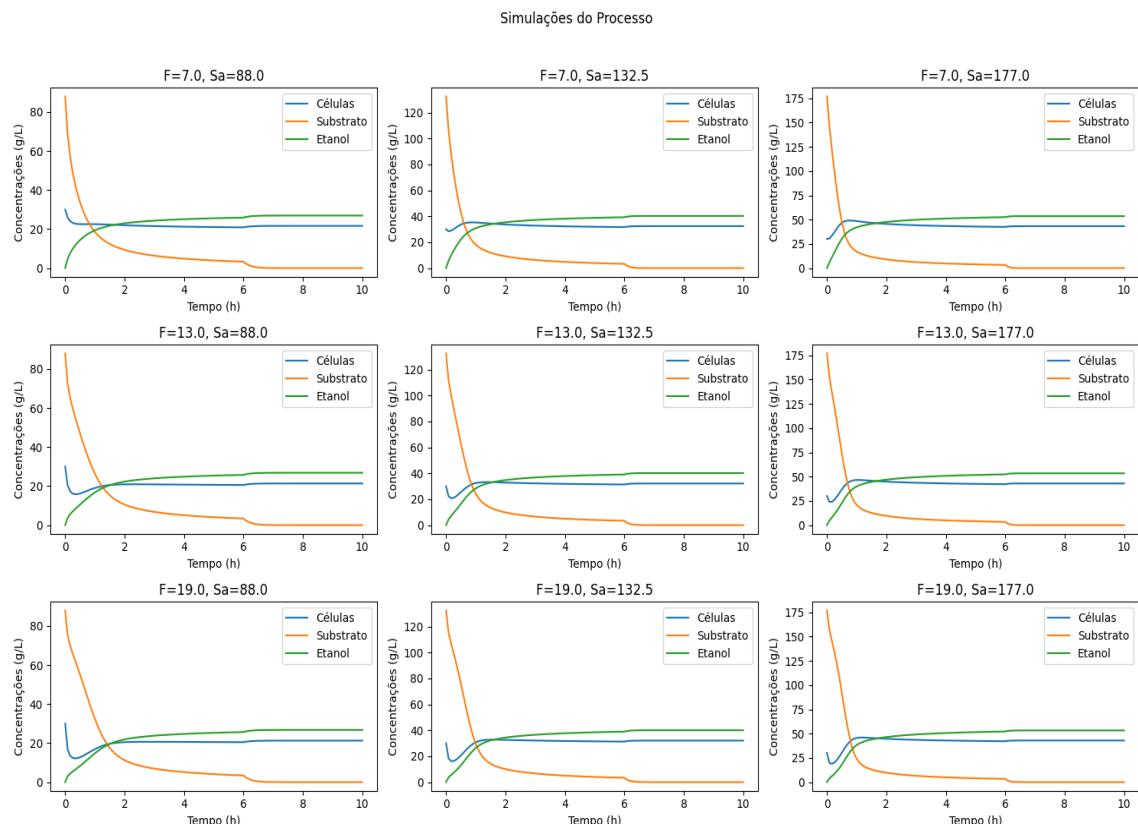
5. RESULTADOS E DISCUSSÃO

Os resultados serão apresentados conforme a ordem cronológica de execução dos códigos. Logo após serão mostrados o comparativo das métricas entre os três modelos, para os dois testes executados.

5.1. Simulação do Processo Fermentativo

As Figuras 12 e 13 exibem os gráficos das simulações feitas para as diversas combinações, conforme citado na Tabela 4. Foi possível verificar que a concentração inicial de substrato elevada fez com que a concentração de células diminua antes de aumentar no início, conforme prevê o modelo de Teissier. Esse fato dificultou os modelos de predição, principalmente para os valores iniciais, conforme será mostrado mais adiante.

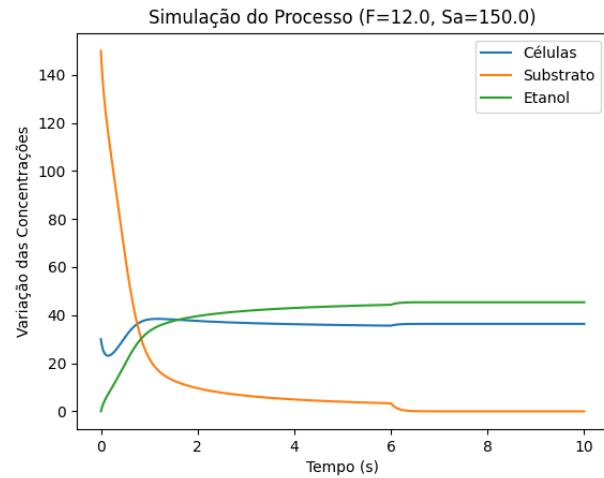
Figura 12 - Gráficos das simulações feitas para geração e coleta de dados do processo.



Fonte: Autor (2024).

Após a interrupção na adição da alimentação é possível ver que há um aumento na concentração de etanol produzido.

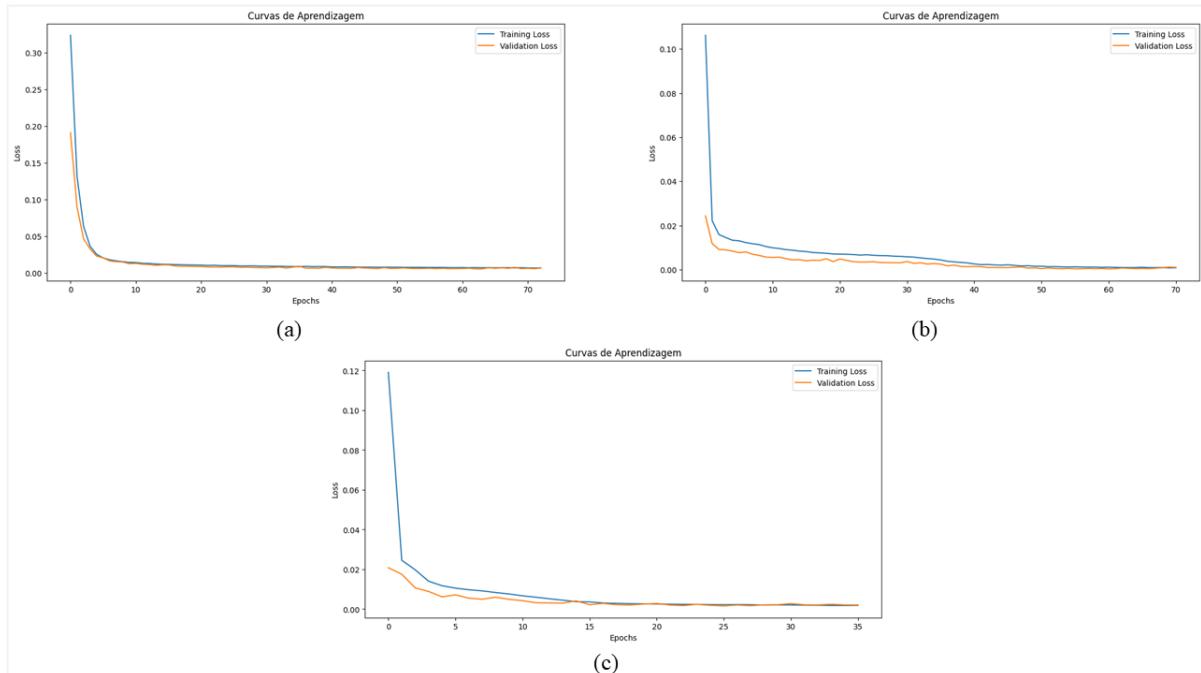
Figura 13 - Gráfico da simulação para geração de dados para o teste dos modelos.



Fonte: Autor (2024).

Durante o treinamento da rede é acompanhado a função de perda (*loss*, em inglês) e ela deve ter um comportamento característico quando um modelo está bem generalizado ou não. Para uma boa generalização, as curvas de perda do treinamento e de validação devem diminuir gradativamente até um ponto de convergência, conforme pode ser visto na Figura 14 para os modelos utilizados neste trabalho.

Figura 14 - Curvas de perda do treinamento e da validação. A figura (a) é a rede MLP, a (b) da rede LSTM e a (c) é do LSTM-AE.



Fonte: Autor (2024).

5.2. MLP

A Tabela 10 exibe os valores das métricas utilizadas para cada uma das variáveis estudadas. É possível ver que o valor de R^2 foi maior e os valores RMSE e MAE foram menores para o etanol, mas o contrário aconteceu com a concentração de células, mostrando assim o quanto a variação na concentração devido ao decaimento celular influencia na capacidade do modelo.

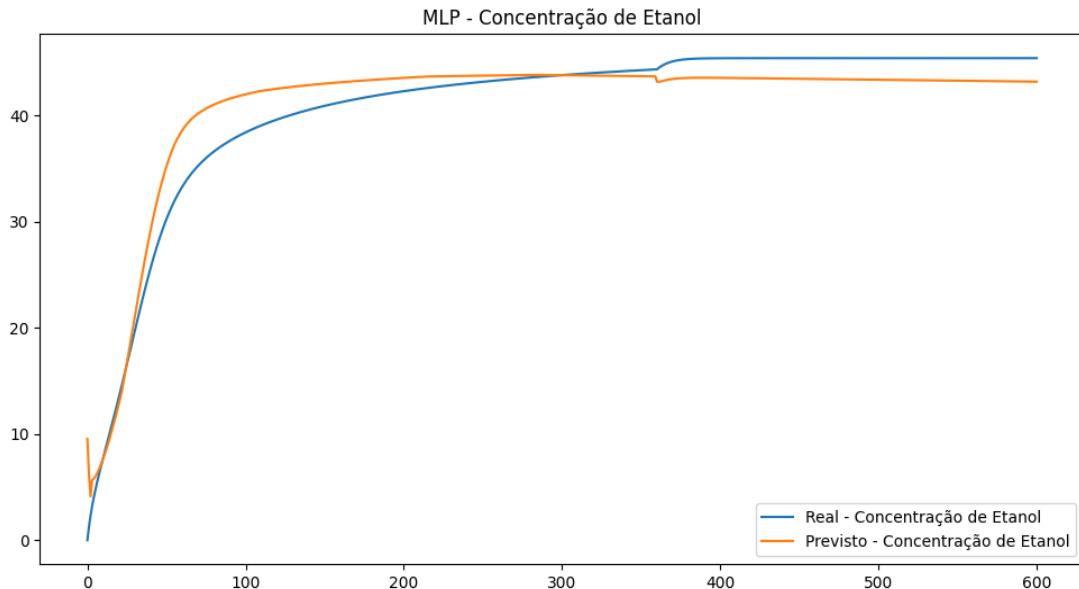
Tabela 10 – Métricas de desempenho da MLP para o primeiro teste.

Variável	R^2	RMSE	MAE
Etanol	0,9285	2,2360	1,7562
Substrato	0,9188	6,7738	2,4014
Células	0,7253	1,3219	0,6938

Fonte: Autor (2024).

No início da simulação o modelo prevê com um erro maior, mas rapidamente ele converge para um valor bem próximo do real, conforme mostra a Figura 15. A maior diferença, embora pequena, está na primeira hora da simulação, quando os valores estão convergindo para um valor estacionário. É possível visualizar também que o modelo prevê o aumento da concentração quando a alimentação cessa.

Figura 15 - Gráfico das concentrações de etanol real versus previsto pela MLP para o primeiro teste.



Fonte: Autor (2024).

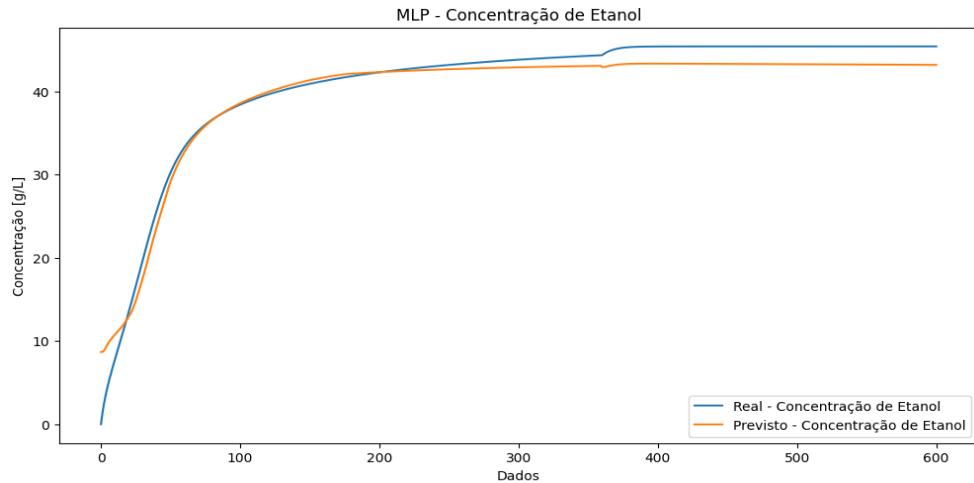
Para o segundo teste, no qual foi utilizado apenas os dados da concentração de etanol para o treinamento da rede, verificou-se um aumento na acuracidade geral do modelo, com um valor de R^2 maior e erros menores, conforme mostra a Tabela 11 e a Figura 16.

Tabela 11 - Métricas de desempenho da MLP para o segundo teste.

Variável	R ²	RMSE	MAE
Etanol	0,9783	1,2326	0,9320

Fonte: Autor (2024).

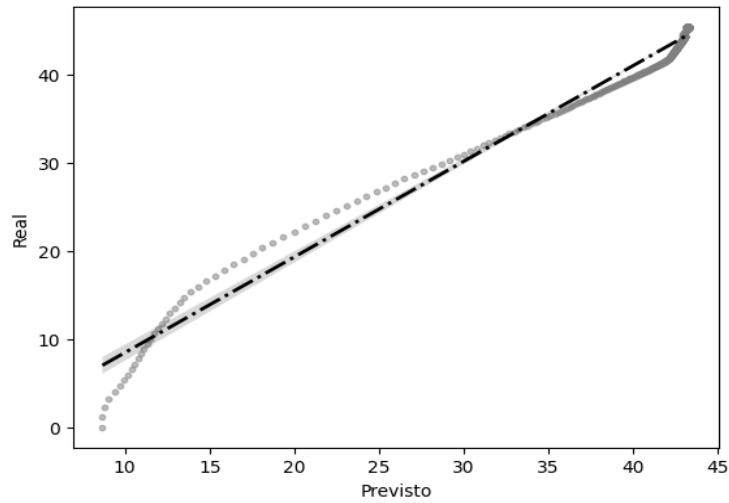
Figura 16 - Gráfico das concentrações de etanol real versus previsto pela MLP para o segundo teste.



Fonte: Autor (2024).

A Figura 17 mostra o gráfico de paridade entre o valor previsto e o real.

Figura 17 - Gráfico de paridade entre os valores reais e previstos da MLP.



Fonte: Autor (2024).

5.3. LSTM

A rede LSTM se mostrou melhor que a MLP de um modo geral, como referenciado na literatura, pois conseguiu desempenhar bem para as três variáveis principais do estudo. Para o

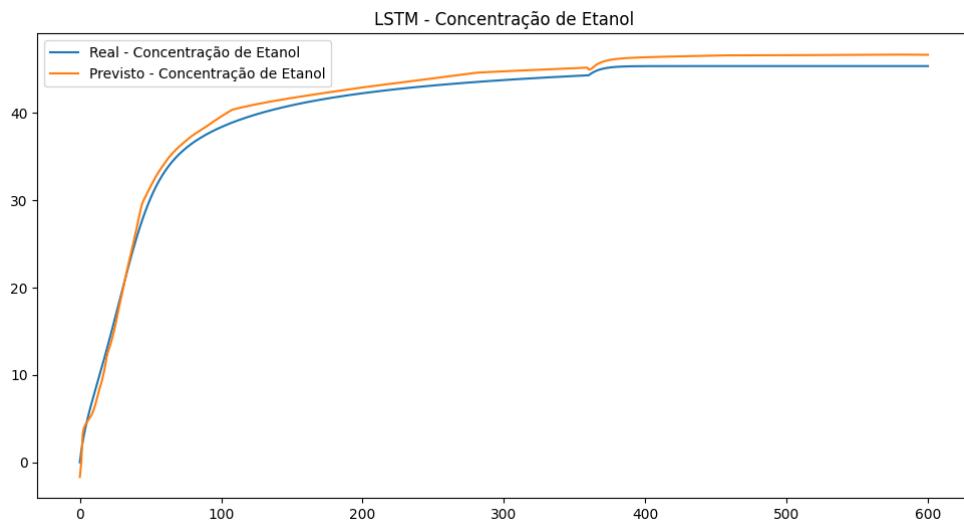
segundo teste da LSTM a rede desempenhou ainda melhor. As Tabelas 12 a Figura 18 mostram os resultados para o primeiro teste.

Tabela 12 - Métricas de desempenho da LSTM para o primeiro teste.

Variável	R ²	RMSE	MAE
Etanol	0,9614	1,6419	1,3915
Substrato	0,9570	4,9307	1,9079
Células	0,8356	1,0227	0,4990

Fonte: Autor (2024).

Figura 18 - Gráfico das concentrações de etanol real versus previsto pela LSTM para o primeiro teste.



Fonte: Autor (2024).

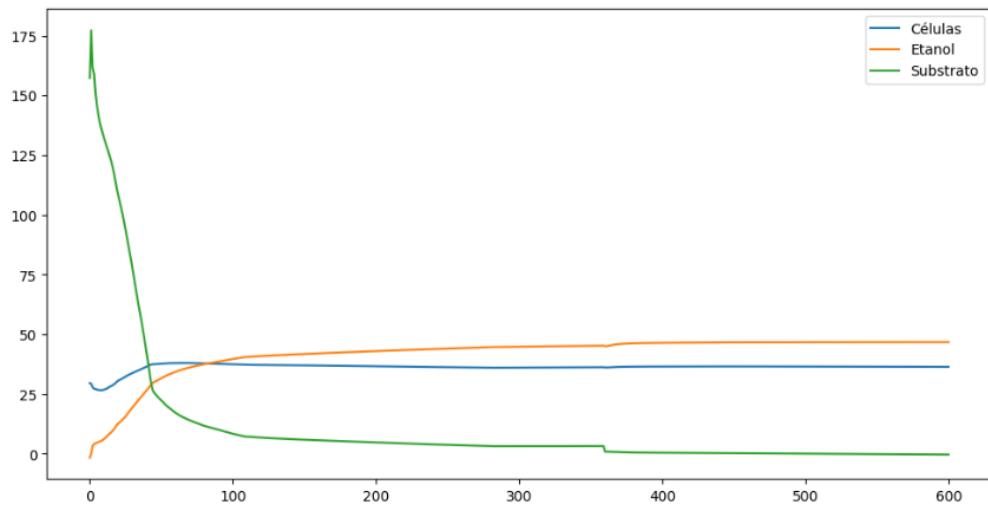
A rede LSTM conseguiu se adaptar bem a variação inicial dos valores, que são maiores. A tabela 13 mostra e as Figuras 19, 20 e 21 mostram os resultados para o segundo teste.

Tabela 13 - Métricas de desempenho da LSTM para o segundo teste.

Variável	R ²	RMSE	MAE
Etanol	0,9973	0,4336	0,3055

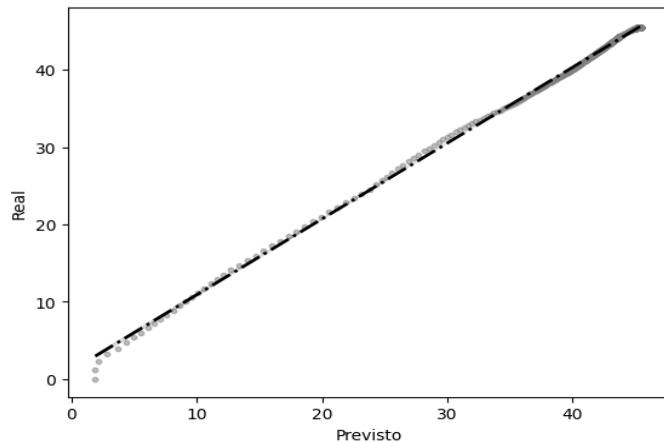
Fonte: Autor (2024).

Figura 19 - Variáveis do processo ao longo da simulação.



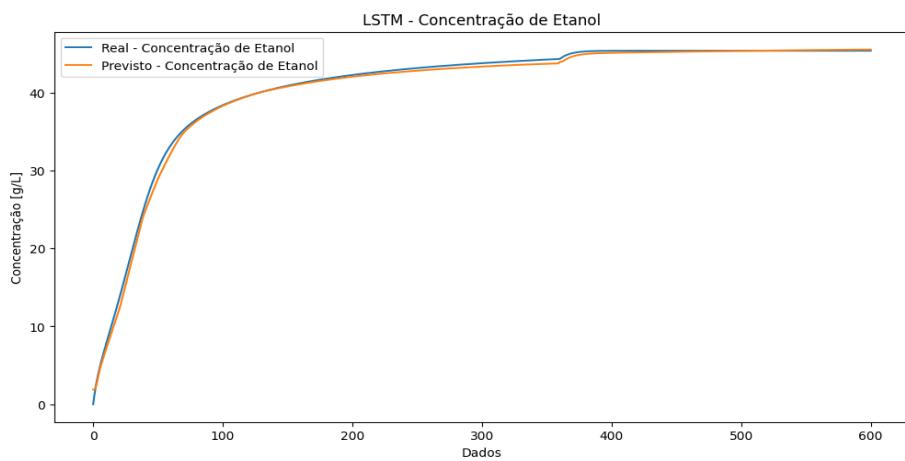
Fonte: Autor (2024).

Figura 20 - Gráfico de paridade entre os valores reais e previstos da LSTM para o segundo teste.



Fonte: Autor (2024).

Figura 21 - Gráfico das concentrações de etanol real versus previsto pela LSTM para o segundo teste.



Fonte: Autor (2024).

5.4. LSTM-AUTOENCODER

Para a simulação de teste, o modelo foi capaz de prever com muita exatidão mesmo utilizando os dados reduzidos, conforme mostram os resultados na Tabela 14. As métricas do teste utilizando apenas os dados de etanol estão na Tabela 15 e, diferente dos outros modelos, o modelo desempenhou um pouco pior em relação a utilização de todas as variáveis.

Tabela 14 - Métricas de desempenho da LSTM-AE para o primeiro teste.

Variável	R ²	RMSE	MAE
Etanol	0,9661	1,5385	1,2371
Substrato	0,9540	5,0969	2,6388
Células	0,7389	1,2888	0,7488

Fonte: Autor (2024).

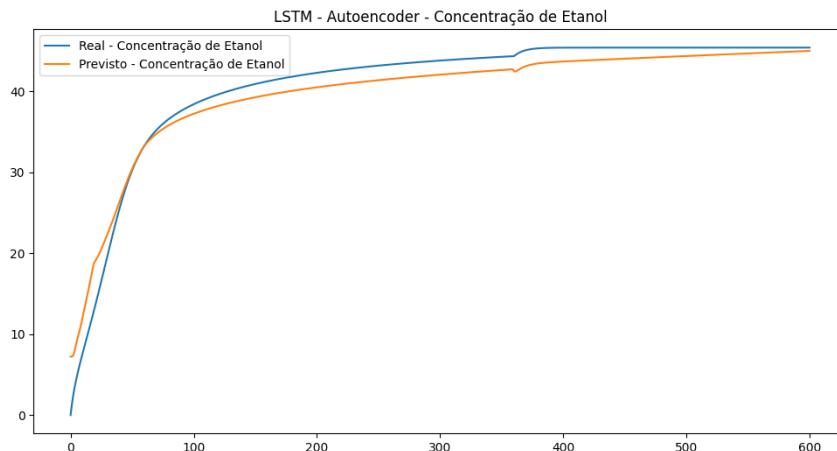
Tabela 15 - Métricas de desempenho da LSTM-AE para o segundo teste.

Variável	R ²	RMSE	MAE
Etanol	0,9551	1,7722	1,5242

Fonte: Autor (2024).

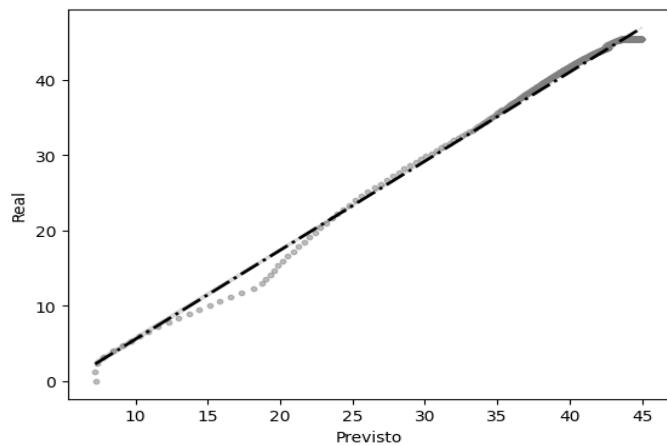
A rede se mostrou bem capaz de realizar inferências mesmo utilizando os dados com dimensionalidade reduzida, quase não perdendo acuracidade quando comparada a LSTM, o que mostra que o método é válido quando se deseja aumentar a eficiência computacional. As Figuras 22 e 23 mostram o gráfico do real versus o previsto e da regressão para o segundo teste.

Figura 22 - Gráfico das concentrações de etanol real versus previsto pela LSTM-AE para o segundo teste.



Fonte: Autor (2024).

Figura 23 - Gráfico de paridade entre os valores reais e previstos da LSTM-AE para o segundo teste.



Fonte: Autor (2024).

6. CONCLUSÃO

No presente trabalho, desenvolveu-se um *soft-sensor* em linguagem Python, utilizando redes neurais para a predição de variáveis do processo de fermentação alcoólica em batelada alimentada. A partir de simulações realizadas, foram gerados dados que permitiram o treinamento e a avaliação dos modelos, atingindo os objetivos propostos com resultados satisfatórios.

A rede LSTM se destacou como a técnica mais eficaz, apresentando melhores métricas para as variáveis avaliadas ($R^2 = 0,9973$, RMSE = 0,4336 e MAE = 0,3055). Já o modelo LSTM-*Autoencoder* mostrou-se uma alternativa promissora, capaz de manter boa precisão mesmo utilizando uma quantidade menor de informação, se mostrando relevante para aplicações onde a eficiência computacional é um ponto crítico. Em contrapartida, a MLP, embora tenha desempenhado de forma adequada, teve o pior resultado e apresentou limitações para capturar a complexidade das dinâmicas do processo.

A aplicação da linguagem Python e suas bibliotecas para simulação do processo e desenvolvimento dos modelos confirmou ser eficaz, permitindo atender os objetivos propostos. Além disso, a integração de conceitos aprendidos durante a graduação nas disciplinas de métodos numéricos, biotecnologia industrial e, principalmente, tópicos de inteligência artificial aplicados a engenharia, mostrou-se crucial para o sucesso deste trabalho, evidenciando a relevância desses conhecimentos para a formação de um engenheiro químico na era da Indústria 4.0.

Embora os resultados tenham sido satisfatórios algumas melhorias podem ser implementadas para aprimorar ainda mais o desempenho e a aplicabilidade do sensor. Uma das possíveis melhorias envolve testar diferentes arquiteturas, alterando o número de camadas, testando diferentes quantidades de neurônios e utilizando diferentes combinações de funções de ativação na arquitetura dos modelos.

Outra possível melhoria seria a ampliação da base de dados. A coleta de dados reais de processos industriais permitiria uma comparação mais robusta dos modelos com cenários mais próximos da aplicação prática. Além disso, aumentar o número de simulações e/ou variar os parâmetros da simulação pode ajudar a tornar os modelos mais precisos.

REFERÊNCIAS

- ABIODUN, O. I. et al. **State-of-the-art in artificial neural network applications: A survey.** Heliyon, vol. 4 (11), 2018. Disponível em: <https://www.sciencedirect.com/science/article/pii/S2405844018332067>. Acesso em: 21 jan. 2024.
- ALIABADI, M. M. **Process data analytics using Deep Learning Techniques.** Dissertação (Mestrado) — Curso de Computer Science, Graduate School, Wayne State University, Detroit, 2020.
- BAILEY, J. E., OLLIS, D. F. **Biochemical Engineering Fundamentals.** 2 ed. New York: McGraw-Hill, 1986. p. 374.
- BANITAAN, S. et al. **A Review on Artificial Intelligence in the Context of Industry 4.0.** International Journal of Advanced Computer Science and Applications, Vol. 14, No. 2, 2023. p. 24-30.
- BELO, M. I. O. **Sistemas Inteligentes Aplicados à Otimização de Parâmetros Cinéticos e Modelagem de Bioprocessos.** Trabalho de Conclusão de Curso – Faculdade de Engenharia Química, Universidade de Alagoas. Maceió, 2021.
- BRASIL. Companhia Nacional de Abastecimento (Conab). Disponível em: <https://www.conab.gov.br/ultimas-noticias/5489-producao-de-cana-de-acucar-na-safra-2023-24-chega-a-713-2-milhoes-de-toneladas-a-maior-da-serie-historica>. Acesso em: 10 nov. 2024.
- BURKOV, A. **The Hundred-Page Machine Learning Book.** Andriy Burkov, 2019.
- CASTRO, L. N. **Fundamentals of natural computing:** basic concepts, algorithms, and applications. Chapman & Hall, 2007.
- COELHO, F. A. et al. **Modelagem do Processo de Fermentação Alcoólica em Batelada Alimentada e Determinação do Perfil Ótimo de Alimentação de Mosto.** VIII COBEQ-IC. Minas Gerais, 2009.
- DESAI, K. et al. **Soft-sensor development for fed-batch bioreactors using support vector regression.** Biochemical Engineering Journal 27 (2006) p. 225–239.
- DOW. **Dow recebe prêmio Future Edge 50 2021 por capacidade de inteligência preditiva.** 27 abr. 2021. Disponível em: <https://br.dow.com/pt-br/noticias/dow-recebe-premio-future-edge-50-2021-por-capacidade-de-inteligencia-preditiva.html>. Acesso em: 10 nov. 2024.
- HAYKIN, S. **Neural Networks: A Comprehensive Foundation.** Delhi: Pearson Education, 1999.
- HOUDT, G. V., MOSQUERA, C., NÁPOLES, G. **A review on the long short-term memory model.** Artificial Intelligence Review, vol. 53 (8), 2020.

- Indústria 4.0: muito além da automação.** Bosch no Brasil. Disponível em: <https://www.bosch.com.br/noticias-e-historias/industria-4-0/>. Acesso em: 12 jan. 2024.
- JÚNIOR, J. B. J. Tecnologia e Fabricação do Álcool.** Inhumas: IFG; Santa Maria: Universidade Federal de Santa Maria, 2012.
- KAY, S. et al. Integrating Autoencoder and Heteroscedastic Noise Neural Networks for the Batch Process Soft-Sensor Design.** Industrial & Engineering Chemistry Research, 2022 61 (36), p. 13559-13569.
- KIM, S. G., CHAE, Y. G., SEONG, P. H. Signal Fault Identification in Nuclear Power Plants based on Deep Neural Networks.** 30th DAAAM International Symposium On Intelligent Manufacturing And Automation, 2019. p. 0846-0852.
- LIMA, J. M. M. Extração de Características Representativas Para o Desenvolvimento de Soft-sensor Industriais: Uma Abordagem Baseada em Aprendizado Profundo.** Tese de Doutorado - Programa de Pós-Graduação em Engenharia Elétrica e de Computação da UFRN. Natal, 2021.
- LIMA, U. A. et al. Processos Fermentativos e Enzimáticos.** 2 ed. São Paulo: Blucher, 2019.
- LOTUFO, F. A., GARCIA, C. Soft-sensor ou Soft Sensors: Uma Introdução.** 7th Brazilian Conference on Dynamics, Control and Applications. FCT – Unesp, São Paulo, 2009.
- MENEGOLLA, H. B. Estudos sobre a aplicação de autoencoder para Construção de Inferências na Indústria Química.** Tese (Mestrado em Engenharia Química) – Universidade Federal do Rio Grande do Sul. Porto Alegre, 2019.
- PERERA, Y. S. et al. The role of artificial intelligence-driven soft sensors in advanced sustainable process industries: A critical review.** *Engineering Applications of Artificial Intelligence*, v. 121, 2023. Disponível em: <https://doi.org/10.1016/j.engappai.2023.105988>. Acesso em: 10 nov. 2024.
- SCHMIDELL, W. et al. Biotecnologia Industrial.** vol. 2. São Paulo: Blucher, 2001.
- SHELL. Artificial intelligence.** Disponível em: <https://www.shell.com/what-we-do/digitalisation/artificial-intelligence.html>. Acesso em: 10 nov. 2024.
- SOARES, F. D. R. Técnicas de Machine Learning Aplicadas a Inferência e Detecção e Diagnóstico de Falhas de Processos Químicos Industriais em Contexto Big Data.** Tese (Mestrado em Tecnologia de Processos Químicos e Bioquímicos) – Universidade Federal do Rio de Janeiro. Rio de Janeiro, 2017.
- TORTORA, G. J.; FUNKE, B. R.; CASE, C. L. Microbiologia.** 10 ed. Porto Alegre: Artmed, 2012. p. 135.

- VASCONCELOS. J. N. **Operação e Simulação do Processo de Fermentação Alcoólica em Batelada Alimentada com Vazão Variação de Alimentação.** Rio de Janeiro, 1987. (Dissertação de Mestrado. Escola de Química da UFRJ).
- VIDAL, M. F. **Agroindústria – Etanol.** Caderno Setorial ETENE. Ano 7, Nº 237, set. 2022. Disponível em: https://www.bnb.gov.br/s482-dspace/bitstream/123456789/1409/3/2022_CDS_237.pdf. Acesso em: 15 jan. 2024.
- WANG, K. et al. **Dynamic Soft Sensor Development Based on Convolutional Neural Networks.** Industrial & Engineering Chemistry Research 2019 58 (26), p. 11521-11531.
- YAN, W., SHAO, G., WANG, X. **Soft sensing modeling based on support vector machine and Bayesian model selection.** Computers and Chemical Engineering 28 (2004). p. 1489–1498.
- ZHU, X. et al. **Modern Soft-Sensing Modeling Methods for Fermentation Processes.** Sensors 2020, 20(6), p. 1771.