UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

ESCOLA DE ENGENHARIA

DEPARTAMENTO DE ENGENHARIA QUÍMICA

ENG07053 - TRABALHO DE DIPLOMAÇÃO EM ENGENHARIA QUÍMICA

Desenvolvimento de um Analisador Virtual de Teor de Sólidos no Processamento de Proteína Isolada de Soja

Autor: André Ruzicki Corsetti

Orientador: Prof. Dr. Marcelo Farenzena

Porto Alegre, dezembro de 16

Sumário

Agradecimentos

Agradeço à minha família, por sempre ter me dado o suporte necessário e todas as condições para eu seguir o meu caminho. O apoio e carinho de vocês são a base de tudo que eu tenho.

À minha companheira, Nina, desde sempre do meu lado mesmo com todos os percalços. Tua parceria é tudo para mim.

Aos meus amigos, que não deixaram eu enlouquecer no meio do percurso, sempre dando estímulo para eu sair um pouco da linha: tenham certeza que isso foi essencial para eu conseguir chegar até aqui. Um salve pra galera do Bonde, da UFRGS e de Manchester.

Aos professores do DEQUI, por sempre estarem disponíveis e engajados com o ensino, mesmo diante de todas as adversidades que encontramos na Universidade em nossos dia-a-dia. Vocês são um exemplo de perseverança.

Ao Professor Farenzena, pela excelente orientação, muita paciência e compreensão.

Muito obrigado!

Resumo

Analisadores virtuais são algoritmos que utilizam um modelo matemático para predição de algum parâmetro de processo usualmente obtido por meio de análises de laboratório ou de instrumentos cuja taxa de amostragem é bastante baixa. Assim, a utilização de um analisador virtual em um processo confere em tempo real a estimativa de um dado antes obtido por meio de amostras significativamente espaçadas, conferindo maior confiabilidade, estabilidade e controle do processo e diminuindo a necessidade de análises laboratoriais. O presente trabalho trata do desenvolvimento de um analisador virtual para inferência do Teor de Sólidos em uma unidade produtora de proteína de soja isolada, baseando-se em modelos de Regressão Linear Multivariável e Redes Neurais Artificiais. A metodologia aplicada mostrou-se eficiente, tendo-se obtido modelos para predição do Teor de Sólidos com valores de R^2 acima de 0,90 em ambos os métodos de modelagem, com erro relativo de menos de 2% em relação aos dados de análises laboratoriais.

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

TS – Teor de Sólidos

TS_i – Teor de Sólidos na entrada da Etapa de Neutralização

TS_o – Teor de Sólidos no Vaso de Diluição

VD – Vaso de Diluição

T_{VD} – Temperatura no Vaso de Diluição

ρ_{VD} – Massa Específica no Vaso de Diluição

L_{TP} – Nível de um Tanque Pulmão Intermediário

T_{TT} – Temperatura do Tratamento Térmico

T_{CF} – Temperatura da Câmara Flash

μ_{CF} – Viscosidade na Câmara Flash

ρ_{CF} – Massa Específica na Câmara Flash

P_{CF} – Pressão na Câmara Flash

L_{CF} – Nível da Câmara Flash

h_i – Vetor de saída da camada i de uma Rede Neural

J_i – Transformação funcional da camada i de uma Rede Neural

W – Soma ponderada das entradas da camada i de uma Rede Neural

S – Função Sigmoidal

u – Variável de entrada da Rede Neural

τ – Tempo morto

Lista de Abreviaturas e Siglas

LMR – Regressão Linear Multivariável

PCA – Análise de Componentes Principais

PLS – Mínimos Quadrados Parciais

ANN – Redes Neurais Artificiais

FANN - Rede Neural Artificial do tipo *feed-forward*

SVM – Máquinas de Suporte Vetorial

SSE – Soma Quadrática do Erro

S_{YY} – Média do conjunto de dados

R^2 – Coeficiente de Determinação

1 Introdução

A enorme competitividade entre as indústrias de processo no atual cenário econômico mundial torna necessário um excelente monitoramento de qualidade de produtos, obtido principalmente pela utilização de um grande volume de sensores na instalação industrial para monitoramento e controle do processo. Apesar disso, inúmeros parâmetros importantes para o controle de qualidade não são possíveis de serem medidos de forma contínua, seja por motivos de inexistência de instrumentação adequada ou por elevado custo na aquisição e manutenção de um sensor específico. Esses parâmetros, portanto, são conhecidos somente através de análises laboratoriais, que muitas vezes são realizadas em uma frequência aquém da desejada e com uma demora significativa na obtenção de tais índices (FACCHIN, 2005).

Neste contexto, pesquisadores vêm desenvolvendo modelos matemáticos que permitam correlacionar o grande volume de dados obtidos em processo para predizer uma variável que não se consegue medir diretamente, ou cuja amostragem é insuficiente para controle automático. Torna-se possível, portanto, inferir variáveis de difícil aquisição de maneira contínua através da utilização de variáveis secundárias do processo e de um modelo matemático adequado (FACCHIN, 2005). Esses algoritmos, no âmbito de processos industriais, são chamados de analisadores virtuais, sensores inferenciais ou sensores baseados no observador (KADLEC *et al.* 2009).

A soja é uma oleaginosa de alto teor proteico, sendo a concentração de proteína no grão de aproximadamente 40% (LIN, 1997). Devido ao seu alto valor nutricional e propriedades funcionais, a proteína de soja é produzida industrialmente nas formas de proteína isolada (concentração maior que 90%) e concentrada (concentração maior que 70%). Estes produtos são utilizados como ingredientes na indústria alimentícia, tanto com o objetivo de aumentar o teor de proteínas do alimento, quanto para oferecer alguma característica específica ao produto, tal como a capacidade de formar emulsões (ex.: salsicha), conferir textura a um produto cárneo (ex.: *hamburger, nuggets*) ou aumentar a capacidade de retenção de água do alimento (ex.: apresuntados, frango, *chester*). A proteína isolada de soja também é vendida na sua forma pura, como um suplemento proteico para ganho de massa muscular. De acordo com a consultoria de mercado MarketsandMarkets, projeta-se que o mercado mundial de ingredientes de proteína de soja atinja o valor de U$ 10,12 bilhões em 2020.

O presente trabalho discorre sobre a utilização de um analisador virtual para inferência do Teor de Sólidos na corrente de alimentação de um secador do tipo *Spray Dryer* em uma planta de produção de proteína de soja isolada. O parâmetro de Teor de Sólidos é importante tanto para garantir que o produto sofra uma atomização adequada na entrada no secador, quanto para base de cálculo para dosagem de aditivos na corrente de produto. Assim, o analisador virtual irá garantir uma maior confiabilidade no valor de Teor de Sólidos, melhorando o controle de dosagem de aditivos e água no processo.

O objetivo deste trabalho é desenvolver um analisador virtual baseado em um modelo caixa-preta para inferência do Teor de Sólidos em uma unidade produtora de proteína isolada. Além disso, deseja-se avaliar as diferentes estratégias de desenvolvimento, comparando-se a utilização de modelos de Regressão Linear Multivariável e modelos não-

lineares de Redes Neurais Artificiais com diferentes configurações e estratégias de treinamento.

O presente trabalho é dividido como segue: no Capítulo 2 a revisão bibliográfica apresenta os analisadores virtuais, com as suas principais características e vantagens, a metodologia a ser seguida durante o seu desenvolvimento e também exemplifica algumas aplicações dos mesmos na indústria de alimentos. O Capítulo 3 divide-se em caso de estudo, no qual a planta industrial para a qual este trabalho é destinado é apresentada, e metodologia, cujo conteúdo descreve as técnicas e estratégias utilizadas no desenvolvimento do analisador virtual. O Capítulo 4 apresenta os resultados e discussões acerca dos mesmo, analisando os desempenhos dos modelos elaborados e suas peculiaridades e comparando as técnicas de modelagem utilizadas e as diferentes estruturas dos modelos de Redes Neurais. O Capítulo 5 encerra este trabalho com as conclusões que puderam ser tiradas no decorrer do mesmo, bem como apresenta as ideias de trabalhos futuros.

2 Revisão Bibliográfica

Este capítulo versará sobre o estado da arte do estudo de analisadores virtuais de processos e das principais técnicas utilizadas no desenvolvimento dos mesmos.

2.1 Analisadores Virtuais

Analisadores virtuais são sistemas que utilizam algoritmos matemáticos para a obtenção de estimativas em tempo real de variáveis que não são medidas por sensores, utilizando para isso as suas correlações com dados disponíveis por medições em processo (FORTUNA *et al.* 2005).

A utilização de analisadores virtuais em plantas industriais vem sendo estudada por diversos autores, e se tornou uma prática comum para suprir a necessidade de medidas de variáveis obtidas por meio de análises infrequentes. Na literatura há uma série de trabalhos acerca da implementação de analisadores virtuais, por exemplo, para estimação do índice de fluidez em um reator de polimerização e concentração de biomassa em fermentador contínuo (THAM *et al.* 1991), concentração de gasolina (C5) na corrente de topo e de butano (C4) na corrente de fundo de uma coluna debutanizadora (FORTUNA *et al.* 2005) e viscosidade da corrente de alimentação de um secador do tipo *spray-dryer* (LIN *et al.* 2009). Outros autores, como FACCHIN (2005) e KADLEC *et al.* (2009), focaram seus trabalhos no desenvolvimento de técnicas de análises multivariáveis para o desenvolvimento de analisadores virtuais.

As vantagens de utilização de analisadores virtuais são muitas, e FORTUNA *et al.* (2005) destacam as seguintes:

- Oferecem uma alternativa de baixo custo em comparação com instrumentação cara;
- Podem trabalhar em paralelo com sensores físicos, fornecendo informação útil na identificação de falhas de instrumentação;
- Podem ser facilmente implementados em *hardware* existente (por exemplo, micro controladores) e calibrados de acordo com as mudanças nos parâmetros do sistema;
- Permitem a estimação de dados em tempo real, sendo vantajosos em comparação com análises lentas (por exemplo, cromatografia gasosa), assim melhorando o desempenho de algoritmos de controle.

Os analisadores virtuais consistem em algoritmos compostos basicamente por três blocos, conforme apresentado na Figura 2.1.

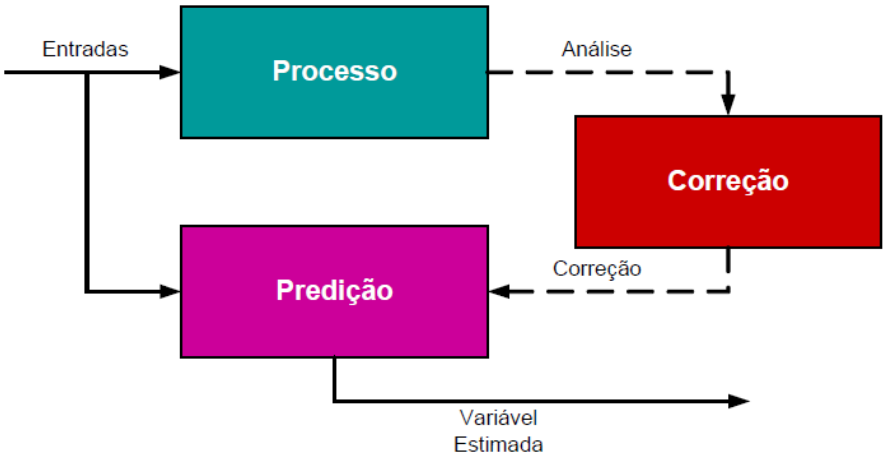

Figura 2.1: Algoritmo básico de um analisador virtual (FACCHIN, 2005).

As entradas do algoritmo consistem nas variáveis secundárias do processo, que são correlacionadas no modelo matemático do analisador virtual, inserido no bloco de *predição*. O bloco *correção* trata de uma estratégia de adaptação para o modelo matemático, baseado na obtenção de dados pela análise laboratorial dos parâmetros de *processo.* Um analisador virtual deve se adaptar ao longo do tempo de acordo com os resultados das medidas laboratoriais as quais se quer substituir, realizando-se uma manutenção contínua no modelo matemático para mantê-lo sempre fiel aos parâmetros mais recentes da planta (FACCHIN, 2005).

De maneira geral, os analisadores virtuais podem ser desenvolvidos com base em modelos fenomenológicos, que levam em conta uma modelagem dos fenômenos físico-químicos do sistema, ou empíricos, cujos modelos são elaborados através de um tratamento matemático sobre os dados de processo (KADLEC *et al.* 2009). O primeiro grupo compreende os chamados modelos caixa-branca, enquanto o segundo os modelos caixa-preta, e cada um apresenta suas características, vantagens e desvantagens, que serão descritas na seção seguinte.

2.2 Técnicas de modelagem

O modelo matemático de um processo consiste em um sistema de equações que, frente o recebimento de dados específicos como variáveis de entrada, fornece como variáveis de saída as respostas do processo. Pode-se subdividir os modelos em três grupos: fenomenológicos, empíricos e análogos. FACCHIN (2005) descreve os tipos de modelos da seguinte forma:

- **Modelos fenomenológicos (ou analíticos)** são aqueles cujas equações são obtidas pela utilização de teorias fundamentadas e princípios básicos da ciência, levando em consideração as leis da física e química e princípios de conservação de massa, energia e quantidade de movimento. O desenvolvimento destes modelos exige um elevado nível de esforço, porém são os que possuem a maior capacidade extrapolativa. São também chamados de modelos caixa branca.

- **Modelos empíricos** não levam em consideração o conhecimento fenomenológico do sistema e suas equações são obtidas pela simples utilização de métodos matemáticos nas variáveis de entrada, relacionando-as com as

variáveis de saída através de técnicas de regressão, por exemplo. Este grupo de modelos apresenta uma baixa capacidade de extrapolação quando comparados com os modelos fenomenológicos, pois as técnicas de regressão servem somente para modelar os dados do sistema, e não o processo como um todo. São também chamados de modelos caixa-preta.

- **Modelos análogos, ou analogias**, são obtidos pela adaptação de um modelo aplicado a um sistema dito análogo, ou seja, com características similares. As equações são replicadas no sistema a ser modelado, com a identificação das variáveis sendo feita por analogia.

- **Modelos semi-empíricos**, também chamados de modelos caixa-cinza, nos quais o conhecimento básico do processo auxilia na determinação de parâmetros desconhecidos, utilizando-se de dados de processo ou experimentais para validação destes parâmetros.

FORTUNA *et al.* (2005) esclarecem que os analisadores virtuais se dividem sumariamente entre os projetados a partir de modelos fenomenológicos e os projetados a partir de modelos empíricos ou semi-empíricos. Conforme descrito anteriormente, a primeira classe requer a utilização dos chamados Modelos de Primeiros Princípios, e por isso exigem um maior entendimento dos processos físico e químico envolvidos no sistema. O autor ainda afirma que, para processos de alta complexidade, a modelagem físico-química do sistema pode ser muito trabalhosa, e muitos parâmetros importantes podem, inclusive, não ser conhecidos. KADLEC *et al.* (2009) descrevem a utilização das técnicas de Filtro de Kalman Estendido e de observador adaptativo para o projeto de analisadores baseados em modelos analíticos.

Já os analisadores virtuais desenvolvidos por modelos empíricos e semi-empíricos são elaborados com base em modelos matemáticos, mesmo que os semi-empíricos tenham uma base parcialmente teórica no seu desenvolvimento. A grande disponibilidade e volume de dados históricos das plantas industriais, que são obtidos para acompanhamento e controle do processo através da utilização de sensores, permite que modelos caixa-preta ou caixa-cinza lineares e não-lineares possam ser elaborados de maneira bastante satisfatória (FORTUNA *et al.* 2005). De acordo com KADLEC *et al.* (2009), as principais técnicas utilizadas no desenvolvimento destes modelos são as de Regressão Linear Multivariável (MLR), Análise de Componentes Principais (PCA) em combinação com um modelo MLR, Mínimos Quadrados Parciais (PLS), Redes Neurais Artificiais (ANN), Sistemas Neuro-Fuzzy e Máquinas de Suporte Vetorial (SVM).

Modelos caixa-preta e caixa-cinza são vantajosos por não demandarem tanto esforço do ponto de vista técnico, visto que o seu desenvolvimento envolve somente a utilização de algoritmos matemáticos para tratamento do grupo de dados de processo. Comparativamente, os modelos caixa-branca muitas vezes requerem balanços de massa e energia, análise da cinética e termodinâmica de reações químicas e outros desenvolvimentos mais complexos. Além disso, os modelos caixa-preta e caixa-cinza representam mais fielmente uma planta industrial já existente, pois utilizam dados reais do processo, ao invés de basearem-se em definições teóricas e ideais (KADLEC *et al.* 2009).

Por outro lado, os autores destacam uma desvantagem na utilização destes modelos devido à sua baixa reprodutibilidade, justamente por carregarem consigo as características

intrínsecas e exclusivas do processo em questão. Os modelos caixa branca são normalmente utilizados em uma etapa de planejamento e concepção do processo, visto que eles descrevem com qualidade o processo em estado estacionário ideal, sem levar em conta as condições reais do processo e as variâncias inerentes a uma planta industrial (KADLEC *et al.* 2009).

2.3 Desenvolvimento de Analisadores Virtuais

O desenvolvimento de analisadores virtuais compreende, de maneira geral, uma série de procedimentos, envolvendo desde a seleção da técnica de modelagem até a estratégia de correção a ser aplicada no modelo desenvolvido (FACCHIN, 2005). Visto que este trabalho propõe o desenvolvimento de um analisador virtual baseado em modelos caixa-preta, este subcapítulo descreverá o procedimento utilizado nestes sistemas. A metodologia apresenta variações na literatura, porém KADLEC *et al.* (2009) destacam que a estrutura básica é essencialmente a mesma, podendo ser aplicada tanto para processos contínuos quanto para processos em bateladas. A Figura 2.2 mostra o procedimento proposto pelos autores.

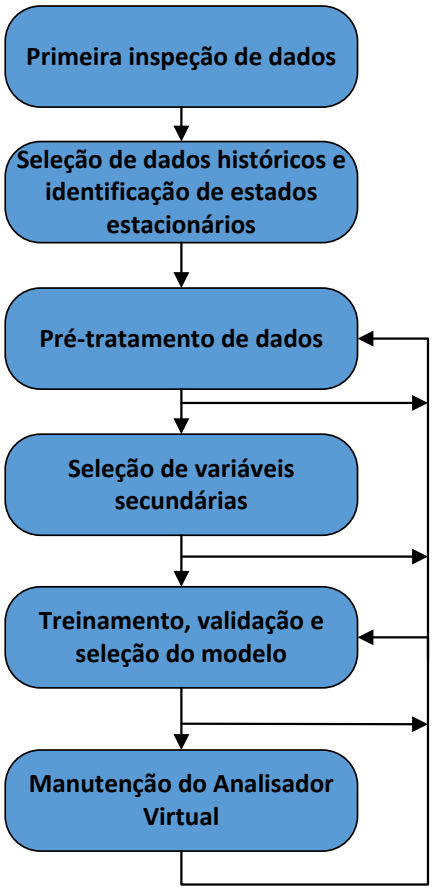

Figura 2.2: Metodologia para desenvolvimento de um analisador virtual (adaptado de KADLEC *et al.,* 2009).

2.3.1 *Primeira Inspeção de Dados*

A primeira etapa no desenvolvimento de um analisador virtual envolve uma inspeção nos dados disponíveis. Deve-se procurar por problemas óbvios, como variáveis com valores congelados por um longo período ou valores não coerentes para variáveis (por exemplo,

valores negativos para dados de vazões), e eliminar do grupo de dados tais valores problemáticos. Nesta etapa, é preciso analisar se a variável de resposta apresenta um comportamento favorável ao desenvolvimento de um modelo que a descreva, sendo necessário que a resposta apresente suficiente variação entre os pontos obtidos para que esta possa ser modelada (KADLEC *et al.* 2009).

2.3.2 *Seleção de Dados Históricos e Identificação de Estados Estacionários*

Os dados utilizados no desenvolvimento do modelo são, na maioria das vezes, adquiridos em estados estacionários do processo. Assim, faz-se necessário uma identificação dos estados estacionários dentro do grupo de dados de processo, selecionando-os para utilização nas etapas de treinamento e avaliação do modelo. No caso de processos em batelada, nos quais o processo está sempre em estado transiente, devem-se utilizar os dados provenientes de bateladas confiáveis e representativas (KADLEC *et al.* 2009).

2.3.3 *Pré-tratamento de Dados*

Esta é uma etapa muito importante no desenvolvimento de analisadores virtuais, pois irá garantir que os dados sejam utilizados de maneira adequada nas etapas subsequentes. A seta retornando para o bloco de Pré-tratamento de Dados na Figura 2.2 demonstra a necessidade de um trabalho continuado neste sentido, sempre levando em conta que os dados devem estar adequados para serem verdadeiramente úteis ao modelo. Dentro dos principais problemas a serem superados neste procedimento, está a presença de dados faltantes e a detecção e substituição de "pontos-fora-da-curva", os chamados *outliers* (KADLEC *et al.* 2009).

2.3.3.1 Dados faltantes

SCHEFFER (2002) destaca que os dados faltantes ocorrem por diversos motivos, e na área industrial costumam ocorrer por problemas na comunicação de sensores, devido à compressão no historiador, ou no *hardware* do historiador de dados. Desta forma, as alternativas para resolver esse problema são a exclusão de todo o grupo de dados referente àquele instante, ou a atribuição de valores para substituição. No caso da atribuição de valores, o autor sugere que as alternativas envolvem a utilização da média aritmética, mediana ou moda da variável num intervalo de tempo próximo ao ponto faltante, a realização de regressões, ou por fim a utilizações de outros algoritmos mais complexos. Essa tomada de decisão requer uma avaliação do impacto que uma má substituição de valores pode trazer, em comparação com a perda de parte do volume de dados para o trabalho, pela exclusão dos dados faltantes.

2.3.3.2 Pontos-fora-da-curva (outliers)

A presença de pontos-fora-da-curva, os *outliers*, segundo QIN (1997), ocorre devido a falhas nos sensores, má interpretação dos resultados de análises de laboratório e outros distúrbios desconhecidos no processo. O autor destaca que os *outliers* podem ser significativos e representarem desvios reais no processo, mas também podem não ter significado verdadeiro, sendo apenas erros de medição. Na prática, estes pontos extremos terão uma influência significativa no resultado do modelo, e por isso devem ser tratados adequadamente.

Os chamados *outliers* óbvios são facilmente identificáveis, por representarem uma situação fisicamente impossível, como, por exemplo, medições de temperatura abaixo do zero absoluto, ou valores de vazão com sinal negativo. Estes dados podem ser substituídos pelo valor médio da variável em um pequeno intervalo de tempo, por valores máximos ou mínimos (o que não é usualmente feito, pelo fato de estes valores não serem facilmente conhecidos), ou ainda tratá-los como dados faltantes, cujo tratamento foi descrito na seção 2.3.3.1 (QIN, 1997).

Os *outliers* não óbvios, por outro lado, são mais dificilmente detectados, pois são dados cujos significados são plausíveis se observados separadamente, mas na realidade fisicamente impossíveis para aquele instante de tempo. Por exemplo, pressão e vazão devem ter uma correlação quadrática de acordo com os princípios físicos, mas um *outlier* pode violar esta relação. De outra perspectiva, um enorme aumento de temperatura em um curto intervalo de tempo, em um sistema cuja dinâmica seja conhecidamente lenta, certamente será um *outlier* e por isso deve ser tratado como tal. O tratamento de *outliers* não óbvios se dá da mesma maneira que com *outliers* óbvios (QIN, 1997).

2.3.4 *Seleção de Variáveis Secundárias*

A capacidade de um modelo matemático de representar fielmente um processo está fortemente ligada à escolha das variáveis secundárias do processo a serem utilizadas na construção do analisador. Segundo FACCHIN (2005), a utilização de variáveis não-sensíveis na elaboração do modelo irá afetar negativamente o seu desempenho, sendo, portanto, a etapa de seleção de variáveis uma das etapas mais críticas no desenvolvimento de um analisador virtual. FORTUNA *et al.* (2005) frisam que o desempenho de modelos não-lineares, como Redes Neurais artificiais, é fortemente impactado pela utilização de variáveis inadequadas.

Para o caso do desenvolvimento de modelos fenomenológicos, a seleção de variáveis se torna relativamente simples, visto que a própria dedução do modelo matemático traz consigo o grupo de variáveis vinculadas ao conjunto de equações. Para modelos empíricos ou semi-empíricos, entretanto, é a seleção de variáveis que irá delinear a forma do modelo matemático, e para tal processo pode-se utilizar diversas técnicas. FACCHIN (2005) destaca as seguintes estratégias para seleção de variáveis:

- **Algoritmos de busca exaustiva**, nos quais se elaboram modelos contendo todas as combinações de variáveis possíveis, comparando-os. Para um sistema com *K* variáveis, e supondo que exista um termo independente para o modelo, o número de modelos gerados pelo método de busca exaustiva será 2^K, o que gera um alto custo computacional para este método;

- **Algoritmos sequenciais**, nos quais se realizam diversas etapas de teste de variáveis, com as variáveis sendo sequencialmente incluídas ou removidas do modelo. Conhecidos como métodos *stepwise*, apresentam três variações na metodologia: pode-se adicionar ao modelo uma variável a cada etapa (seleção por adição), partir-se com um modelo com todas as variáveis e remover uma variável de cada vez (seleção por remoção), ou realizar um procedimento de verificação contra redundância das variáveis a cada nova variável incluída no modelo (*stepwise regression*). No caso da seleção por adição, a cada etapa o algoritmo testa o efeito da adição de cada uma das variáveis ao modelo,

mantendo aquela que ocasionar uma maior diminuição no erro de predição. O procedimento se repete até que a adição de uma variável não cause uma melhora significativa ao modelo. No *stepwise* com seleção por remoção, o procedimento é oposto: parte-se com o modelo com todas as variáveis disponíveis e a cada etapa elimina-se do modelo a variável cuja ausência não ocasionar uma piora no desempenho do modelo. Já na *stepwise regression*, cada etapa consiste em uma seleção por adição seguida de uma seleção por remoção, garantindo que são mantidos no modelo somente variáveis com efeitos únicos, sem redundâncias;

- **Algoritmos de busca aleatória**, nos quais se utilizam, por exemplo, métodos baseados em algoritmos estocásticos. Nesta concepção, um algoritmo de otimização (p.ex. algoritmo genético) gera diversas combinações de variáveis, as quais são evoluídas ao longo das iterações.

O procedimento para a seleção de variáveis se inicia com a separação do grupo de dados em, no mínimo, dois subgrupos: um a ser utilizado para treinamento do modelo, ou seja, para o desenvolvimento propriamente dito do modelo, e o outro para validação do modelo obtido. Assim, os dados de treinamento são submetidos ao algoritmo escolhido e avalia-se a correlação das variáveis de entrada com a variável de resposta, mantendo-se no grupo de dados apenas aquelas variáveis que apresentarem uma correlação significativa. O grupo de dados de validação não é utilizado até que se obtenha um modelo, e, nas etapas subsequentes, testa-se o modelo obtido com um grupo de dados novos para avaliar o seu desempenho (KADLEC *et al.* 2009).

2.3.5 *Seleção, treinamento e validação do modelo*

A etapa mais importante no desenvolvimento de um analisador virtual é, segundo KADLEC *et al.* (2009), a da seleção, treinamento e validação do modelo, pois ela será diretamente responsável pelo desempenho do sistema. Os autores esclarecem que não existe uma maneira teórica única para a determinação do tipo de modelo a ser utilizado, e que esta escolha irá depender bastante das experiências prévias do desenvolvedor. Comumente, inicia-se desenvolvendo um modelo mais simples, como uma regressão linear multivariável, e gradualmente aumenta-se o nível de complexidade do modelo para avaliação da melhora da performance do analisador virtual, caso necessário.

KADLEC *et al.* (2009) expõem algumas metodologias a serem seguidas para a validação do modelo obtido, que irão depender das características do sistema modelado:

- Para os sistemas com um grande volume de dados, o desempenho dos modelos desenvolvidos é avaliado com a utilização da base de dados de validação, conforme descrito na seção 2.3.4.

- Algumas unidades industriais não possibilitam a obtenção de um grande volume de dados. Nesses casos, é vantajoso utilizar alguma ferramenta matemática para melhor aproveitamento dos dados disponíveis, como a técnica de Validação Cruzada. Outras técnicas de re-amostragem estatística, como Agregação *Bootstrap* (*bagging) e Boosting*, criam de maneira aleatória grupos menores de dados, a partir da base de dados de processo. A validação

de cada um dos modelos obtidos é feita com um grupo de dados exclusivo, comparando-se posteriormente os seus desempenhos.

As principais técnicas para avaliação do desempenho do modelo desenvolvido envolvem a utilização de índices de ajuste, usando a soma quadrática do erro (SSE) e a distância entre cada ponto e a média do conjunto de dados (S_{YY}) para cálculo do coeficiente de determinação, comumente chamado de R^2. Os modelos obtidos tem os seus índices comparados, buscando-se aqueles que apresentam os menores valores de SSE e, consequentemente, um valor de R^2 mais próximo de 1. As Equações 2.1, 2.2 e 2.3 demonstram o cálculo destes parâmetros (FACCHIN, 2005).

$$SSE = \sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.1}$$

$$S_{YY} = \sum_{i=1}^{N}(y_i - \bar{y})^2 \tag{2.2}$$

$$R^2 = 1 - \frac{SSE}{S_{YY}} \tag{2.3}$$

Onde y_i são os valores gerados pelo modelo, $\hat{y}_i$ os dados originais utilizados para elaboração do modelo e $\bar{y}$ o valor médio do conjunto de dados originais.

Outra alternativa é a realização de uma análise visual dos resultados obtidos, na qual se observa graficamente a diferença entre os valores preditos e os valores corretos. Este tipo de análise não é indicado pelos autores, visto que ele está sujeito a um julgamento subjetivo do desenvolvedor do modelo (KADLEC *et al.* 2009).

Diversos autores aplicam métodos de agregação para melhorar o desempenho geral do analisador virtual desenvolvido. KUNCHEVA (2004) apresenta diversas variações de métodos de agregação, sendo que a ideia por trás de todos é a de se desenvolver mais de um modelo matemático para o analisador virtual, e combinar as respostas obtidas em cada um para chegar ao valor final de predição. GABRYS (2004) descreve diferentes formas de realizar a combinação de modelos, e analisa em que sistemas é vantajoso utilizar essa estratégia ao invés de um modelo único.

2.3.6 *Manutenção do Analisador Virtual*

As correções do modelo fazem-se necessárias devido ao fato de as características inerentes ao processo variarem com o passar do tempo. Isto faz com que o processo modifique-se com o tempo e o modelo perca precisão em sua capacidade de previsão, apresentando uma diferença entre o valor predito e o valor real. Assim, FACCHIN (2005) afirma que se torna interessante que o modelo matemático passe por um processo constante de adaptação, mantendo-se fiel ao processo, para garantir um alto grau de confiança no analisador virtual.

KADLEC *et al.* (2009) destacam que essas variações podem ter causas distintas. O primeiro grupo compreende as variações causadas por abrasão nos elementos mecânicos da planta industrial, o que leva a uma modificação das condições de processo, como, por exemplo, uma diminuição de vazão causada por desgaste de uma bomba. O outro grupo refere-se a influências externas ao processo, como as variações climáticas, pureza das matérias primas e desativação catalítica, e igualmente irão impactar os parâmetros de processo com o passar do tempo.

A estratégia mais simples para correção de modelos consiste na adição de um *bias* contendo a diferença entre a variável predita e a variável real, cujo valor é obtido por análise laboratorial. Para sistemas com uma frequência alta de análises de laboratório, essa estratégia é bem favorável e eficiente, porém no caso de um grande intervalo entre análises, a adição de um *bias* pode prejudicar o desempenho do analisador virtual ao acrescentar uma diferença que pode não ser verdadeira. Outras estratégias, mais sofisticadas, tratam de alterar os parâmetros do modelo à medida que os resultados de análises laboratoriais são obtidos, ou seja, o modelo é constantemente modificado, ajustando-se aos novos valores da variável de resposta. Destaca-se o uso de métodos de Mínimos Quadrados Recursivos, Filtro de Kalman Estendido e outros métodos que utilizam janelas móveis para essa finalidade (FACCHIN, 2005).

2.4 Analisadores Virtuais e Otimização na Indústria de Alimentos

A indústria de alimentos apresenta um enorme potencial para ferramentas computacionais de engenharia, sendo possível aplicá-las desde o desenvolvimento de projetos conceituais até a operação dos sistemas produtores. As diversas alternativas oferecidas pelos computadores permitem que os processos tornem-se mais eficientes, desenvolvendo-se as fórmulas ótimas para produção em larga escala, em um processo projetado da maneira mais adequada e controlado de modo a otimizar a produção. Destaca-se o uso de recursos de Inteligência Artificial para essa última finalidade (HAVLIK *et al.* 1989).

Algoritmos de otimização compõem a principal ferramenta para melhorias no processamento de alimentos. Uma otimização consiste na escolha da melhor alternativa, dentre as possíveis, para de maneira eficiente e sistemática alcançar-se um fim desejado e definido por uma função objetiva. O uso da técnica de Modelagem de Superfícies de Resposta (RMS) é a principal ferramenta de otimização na indústria de alimentos, mas atenta-se para o enorme potencial de métodos de otimização baseados em modelos rigorosos, que descrevem o processo com precisão e preveem variações temporais de processo (BANGA *et al.* 2003).

Dentre os trabalhos envolvendo modelos rigorosos para otimização disponíveis na literatura, existem diversos sobre a indústria de alimentos. CHEN *et al.* (2001) utilizaram modelos de Redes Neurais Artificiais para predição de variações de parâmetros de qualidade do processo de secagem de mirtilo por osmose reversa, obtendo resultados favoráveis na otimização de cinco variáveis importantes para o processo em questão. GOVAERTS *et al.* (1994) descreveram um sistema de secagem em um secador do tipo *spray dryer* através de modelos dinâmicos mecanísticos, implementando um controle do tipo LQG com uso de dados amostrais para correção de *bias*, permitindo a otimização do controle do teor de umidade do pó. Apesar de esse trabalho não ter sido aplicado diretamente à indústria de alimentos, pode-se replicar a metodologia para processos de secagem de alimentos em *spray dryer*, como é o caso da produção de café solúvel, leite em pó e proteína de soja, visto que a modelagem desenvolvida restringiu-se aos fenômenos de transferência de calor e massa envolvidos no secador.

EERIKÄINEN *et al.* (1993) apresentam diversas aplicações do uso de lógica Fuzzy e Redes Neurais para otimização de processos e desenvolvimento de analisadores virtuais na indústria de alimentos. Modelos de lógicas Fuzzy são utilizados principalmente para sistemas com informações vagas e incompletas, e a aplicação de Redes Neurais se dá para

sistemas nos quais não se possui um modelo matemático exato, valendo-se da capacidade de aprendizado que estes modelos possuem. Destaca-se o uso de lógicas Fuzzy nos trabalhos de DOHNAL (1985) para a modelagem de um fermentador simples, FILEV *et al.* (1985) para a simulação de bateladas de fermentação de etanol, NAKAMURA *et al.* (1985) no controle da fermentação de ácido glutâmico e TURUNEN *et al.* (1985) na aplicação em um processo contínuo com biocatalisadores imobilizados (fermento *invertase*). Já o uso de Redes Neurais foi aplicado com sucesso por LINKO e ZHU (1992) na estimação e predição multiestágio para controle de fermentação com glucoamilase e fermento natural na produção de pães, por CHTOUROU *et al.* (1993) no desenvolvimento de um controlador para processos contínuos de fermentação em tanques agitados e por SIMUTIS *et al.* (1993) na estimação e predição da fermentação de álcool etílico na produção de cerveja.

No âmbito de analisadores virtuais, MONTAGUE *et al.* (1992) realizaram uma ampla discussão sobre o desenvolvimento destas ferramentas para uso em bioprocessos, comparando o uso de modelos lineares adaptativos e modelos de Redes Neurais não-lineares. De maneira prática, BRUWER *et al.* (2007) utilizaram os dados de um sensor vibracional para predição em tempo real da textura de alimentos para lanches industrializados (batatas fritas, biscoitos, etc.), aplicando a técnica de Mínimos Quadrados Parciais para a modelagem do problema e desenhando uma estratégia de controle *feedback* para a variável de resposta. Já LIN *et al.* (2009) desenvolveram um analisador virtual para a viscosidade na corrente de alimentação de leite em um *spray dryer*, correlacionando-a com os valores de densidade da corrente (medida em linha) para auxiliar no controle do tamanho de partículas formadas na atomização na entrada do secador. Por fim, OSORIO *et al.* (2008) inferiram a concentração de etanol durante a destilação do vinho, no processo de produção da bebida “Pisco”. Os autores relacionaram quatro medições de temperatura no destilador, obtendo um bom modelo para o analisador virtual e assim eliminando a necessidade de sensores físicos de etanol no processo.

3 Caso de Estudo e Metodologia

Este capítulo é dividido em duas seções, sendo a primeira dedicada a descrever a planta industrial produtora de proteína de soja isolada. Detalhou-se a parte do processo que envolve a necessidade do conhecimento do Teor de Sólidos, para o qual este trabalho objetiva a utilização de um analisador virtual. Na sequência, a segunda seção descreve a metodologia utilizada no desenvolvimento do analisador virtual, detalhando a implementação da estratégia descrita no Capítulo 2, bem como explicando as técnicas de modelagem de Regressão Linear Multivariável e Redes Neurais Artificiais utilizadas nos modelos elaborados.

3.1 Caso de Estudo – Descrição da Unidade Industrial

O grão de soja é composto, em base seca, por cerca de 40% de proteína, 20% de óleo, 5% de minerais, também chamados de cinzas, e 35% de carboidratos, sendo que aproximadamente 5% destes são referentes a fibras (LIN, 1997).

O processo de produção de proteína de soja isolada na indústria começa com o recebimento da soja, preparação dos grãos e processo de extração do óleo, a partir do qual se obtém o farelo de soja, caracterizado por apresentar no máximo 2% de óleo em sua composição e um alto teor de proteínas e carboidratos. O farelo de soja é moído e produz-se a farinha de soja, que é processada na planta produtora de proteína de soja isolada.

As substâncias que compõem a farinha de soja apresentam diferentes faixas de solubilidade em água, de acordo com o pH do meio: a maior parte dos carboidratos de cadeias curtas (açúcares) é solúvel em meios ácidos e básicos; as proteínas apresentam um ponto isoelétrico entre 4 e 5,5, tornando-se insolúveis neste intervalo de pH; a fração de fibras, por sua vez, é insolúvel em soluções com pH acima de 3 (SGARBIERI, 1996). Assim, o processo de produção de proteína isolada de soja consiste fundamentalmente em alterações de pH do meio, com o intuito de solubilizar ou insolubilizar diferentes frações da farinha de soja, para que seja possível a separação por meios físicos das fases sólida, contendo as moléculas insolúveis naquela etapa, e líquida, contendo as moléculas solúveis. Usualmente, utilizam-se centrífugas contínuas para a separação das fases.

O processo é separado em três seções: etapa úmida, etapa de neutralização e aditivação e etapa de secagem. A etapa úmida, ilustrada na Figura 3.1, começa com a extração das proteínas e carboidratos da farinha de soja em meio alcalino, obtendo-se o chamado licor alcalino na fase solúvel e a fibra de soja na fase sólida. O licor alcalino (rico em proteínas) é a fração de interesse e é direcionado para a etapa seguinte, chamada de precipitação ácida. Nesta etapa, o licor é acidificado para que as proteínas se tornem insolúveis e precipitem para a fase sólida, chamada pasta ácida, e os carboidratos mantenham-se na fase líquida, constituindo o soro ácido, efluente do processo. A pasta ácida é direcionada para tanques pulmão, a partir dos quais se alimenta a etapa de neutralização e aditivação do processo.

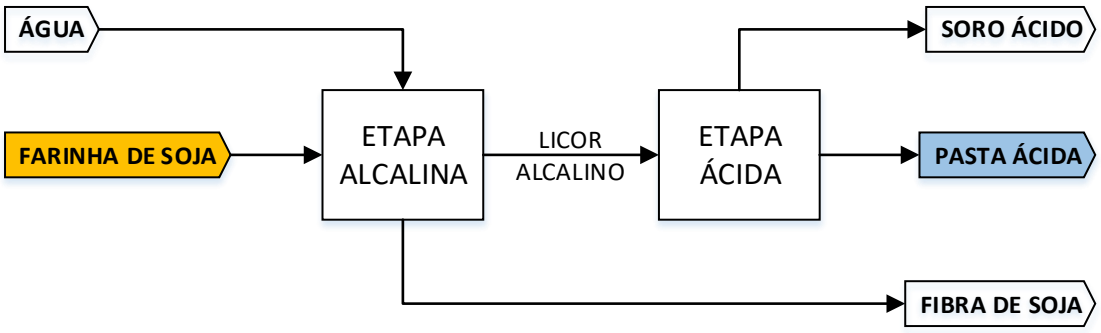

Figura 3.1: Esquema simplificado da etapa úmida do processo.

A etapa de neutralização e aditivação é responsável por conferir a maioria das propriedades físico-químicas do produto final. As diferentes classes de produto devem ser neutralizadas em pHs diferentes e receber os aditivos (minerais e enzimas) de acordo com a sua aplicação final nos alimentos, sendo que um dos parâmetros de maior importância para controle deste processo é o teor de sólidos no vaso de diluição, na etapa de neutralização e aditivação. Este parâmetro de processo possibilita um cálculo exato da quantidade de produto em base seca, permitindo que a dosagem de químicos seja feita de maneira precisa durante a aditivação, o que irá impactar em um melhor controle das características finais do produto. Após a aditivação, a suspensão passa por um tratamento térmico para eliminação de micro-organismos e é armazenado em uma câmara Flash, de onde é bombeado por bombas de alta pressão para um secador do tipo Spray Dryer. O teor de sólidos também é responsável, juntamente com a sua viscosidade, por garantir uma atomização adequada na entrada do secador, formando gotículas de diâmetros ideais e levando a uma secagem adequada. Esta etapa do processo é ilustrada na Figura 3.2 de uma maneira simplificada, mas com detalhes sobre o controle do teor de sólidos no VD.

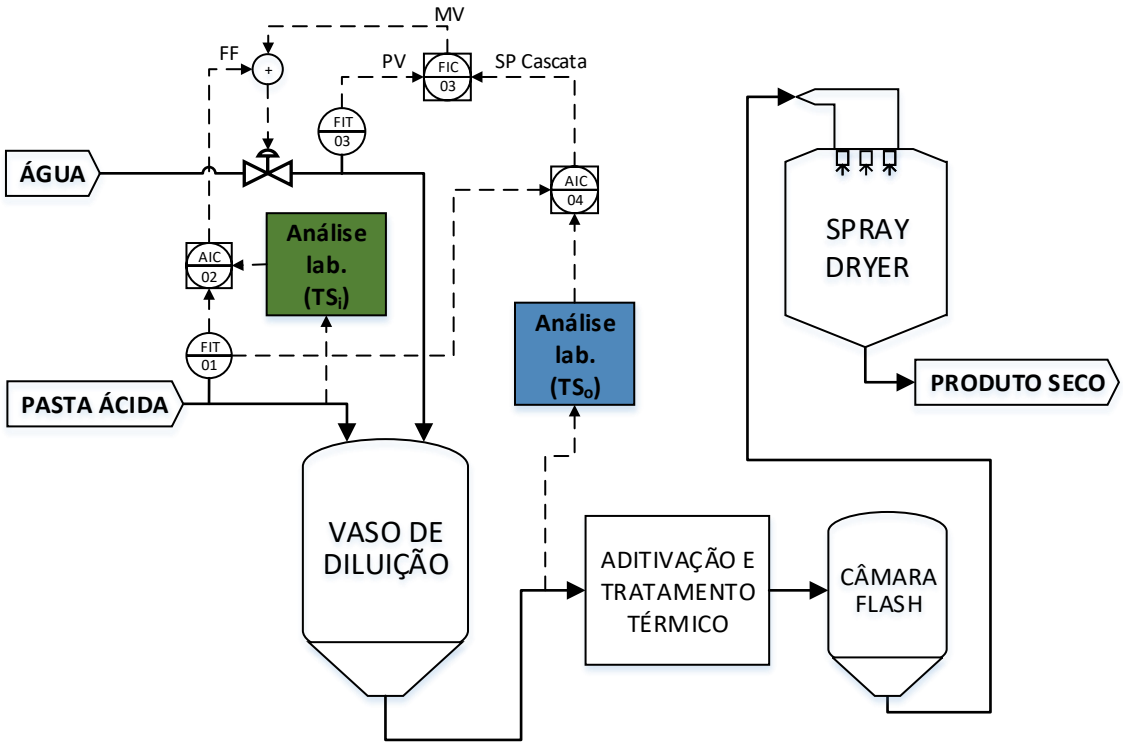

Figura 3.2: Fluxograma simplificado das etapas finais do processo.

O teor de sólidos é controlado no processo atual através da adição de água em um vaso de diluição, tendo como base os valores de teores de sólidos da pasta proteica na entrada da etapa de neutralização ($\mathrm{TS_i}$) e após o vaso de diluição ($\mathrm{TS_o}$), obtidos através de medições laboratoriais realizadas de hora em hora. O valor $\mathrm{TS_i}$ é utilizado para controle do tipo feedforward através do controlador AIC02, somando-se ao valor obtido através do FIC03, que recebe o valor de vazão de água como variável de processo e o compara com o setpoint cascata obtido pelo controlador AIC04, que utiliza o dado $\mathrm{TS_o}$. Vale destacar que os dois controladores de teor de sólidos (AIC02 e AIC04) utilizam o valor de vazão de pasta na alimentação da etapa de neutralização (FIT01), bem como o setpoint de teor de sólidos inserido pelo operador para o determinado produto.

Grande imprecisão é conferida ao teor de sólidos TS_o entre as medições de laboratório, visto que, durante este intervalo, qualquer distúrbio na etapa úmida irá alterar a qualidade de separação da pasta proteica, modificando os valores reais de teor de sólidos na alimentação da etapa de neutralização, os quais não são considerados pela malha de controle de dosagem de água de diluição.

Este trabalho pretende oferecer condições para identificação em tempo real dos distúrbios de processo, utilizando para isso uma inferência do Teor de Sólidos ($\mathrm{TS_o}$) obtida por um analisador virtual nos controladores da etapa de neutralização, a ser corroborada pelos dados obtidos por medições de laboratório. Esta estratégia irá conferir maior rapidez na identificação de variações de processo e consequentemente permitirá um maior controle do parâmetro de Teor de Sólidos na etapa de neutralização e aditivação, garantindo uma boa atomização do produto na alimentação do secador.

3.2 Desenvolvimento do analisador virtual

O desenvolvimento do analisador virtual foi realizado seguindo o procedimento sugerido por KADLEC *et al.* (2009), apresentado no Capítulo de Revisão Bibliográfica. Nesta seção, serão descritas as metodologias utilizadas para implementação de tal analisador.

3.2.1 *Aquisição de Dados de Processo*

A primeira etapa consistiu na obtenção de dados operacionais do processo, tanto das variáveis medidas por instrumentos nos pontos próximos ao Vaso de Diluição (VD), quanto da variável de resposta, o Teor de Sólidos do produto (TS), cujo valor é obtido por análise laboratorial. Os dados de processo coletados envolveram, além do Teor de Sólidos, as seguintes variáveis medidas por sensores: temperatura no VD (T_{VD}), massa específica na saída do VD (ρ_{VD}), nível de um tanque pulmão intermediário (L_{TP}), temperatura do tratamento térmico (T_{TT}), temperatura na Câmara Flash (T_{CF}), viscosidade na Câmara Flash (μ_{CF}), massa específica na Câmara Flash (ρ_{CF}), pressão na Câmara Flash (P_{CF}) e nível da Câmara Flash (L_{CF}). Essas variáveis foram selecionadas primeiramente por acreditar-se que, dentre as medições disponíveis, são as de maior relevância para o Teor de Sólidos do produto na etapa de neutralização.

Conforme mencionado no Capítulo 2, inicialmente é necessário realizar uma inspeção nos dados disponíveis, identificando os estados estacionários do processo e tentando encontrar potenciais problemas na utilização dos dados no desenvolvimento do trabalho. Visto que a unidade em questão lida com material orgânico (soja) com fins de consumo humano, as campanhas de produção são relativamente curtas, de cerca de três dias, com

parada do processo para limpeza das instalações ao final da campanha. Para a coleta de dados, portanto, foi necessário identificar os momentos de partida e parada da planta, garantindo-se que a informação adquirida representasse somente o processo em estado estacionário, durante o período da campanha no qual o processo é estável. Assim, foram eliminados os dados referentes às primeiras 4 horas e às 4 últimas horas de cada campanha. Além disso, procurou-se por distúrbios relevantes no processo, que desviassem o processo de um estado estacionário. Através de uma identificação qualitativa destes desvios, foram eliminados os dados discrepantes.

Os dados das variáveis de processo medidas por sensores foram coletados através do historiador de dados de processo Aspen InfoPlus.21, utilizando-se do *plugin* da ferramenta para o Microsoft Excel. Este *plugin* permite que o intervalo de aquisição dos dados seja definido pelo usuário, bem como o regime de coleta de dados – os valores exportados para o Microsoft Excel podem ser interpolados para o instante de tempo exato a partir dos dados disponíveis mais próximos ao instante em questão, ou os valores exportados podem representar a média do intervalo de aquisição de dados definido pelo usuário. Já os valores de Teor de Sólidos não estão disponíveis no historiador de dados, sendo apenas registrados pela operação em planilhas físicas de controle de processo - são obtidos de hora em hora e representam a medição realizada no instante da coleta de amostra para análise. Por conseguinte, apenas os dados do historiador referentes aos instantes de tempo nos quais as amostras são coletadas para análise são válidos para utilização no trabalho, já que são os momentos que apresentam a totalidade de informações sobre o processo. Assim, decidiu-se pela coleta de dados do historiador com um intervalo curto de tempo – a cada 1 minuto – no regime de coleta de valores médios, e posteriormente realizou-se o cruzamento dos dados com os valores das medições realizadas em laboratório, mantendo-se apenas os instantes de tempo que contivessem informações sobre todas as variáveis do processo. Os instantes para os quais não haviam medições de Teor de Sólidos (59 pontos a cada 60 adquiridos) foram descartados, por não apresentarem utilidade para esse trabalho.

3.2.2 *Pré-tratamento de dados*

O pré-tratamento de dados foi realizado com os dados de processo exportados para o Microsoft Excel®. O principal objetivo desta etapa foi a identificação de *outliers,* além da percepção de dados faltantes. Conforme a bibliografia indica, existem duas alternativas para tratamento destes dados: eliminação dos dados ou substituição por valores médios. Tendo em vista que os dados utilizados no trabalho representavam um instante do processo a cada hora, foi decidido eliminar o conjunto de dados de qualquer instante no qual alguma das variáveis apresentasse a característica de *outlier* ou estivesse faltando, pois a sua substituição muitas vezes não seria confiável ou até mesmo possível (no caso do Teor de Sólidos).

3.2.3 *Segmentação da base de dados*

A partir deste ponto, foi utilizado o software Matlab 7.8.0 (R2009a) para o desenvolvimento do trabalho. Nesta etapa, segmentaram-se os dados em dois grupos – um para treinamento do modelo e outro para validação, conforme indicado pela bibliografia (KADLEC *et al.* 2009). Para tal, utilizou-se a função ***dividerand*** do Matlab, que realiza uma seleção aleatória dos dados e os agrupa em até três novas matrizes. Optou-se por segmentar os dados em apenas dois grupos, sendo que 80% foram destinados aos dados de treinamento e 20% aos dados de validação.

3.2.4 *Seleção de variáveis secundárias*

A seleção das variáveis secundárias foi realizada utilizando-se um algoritmo sequencial do tipo stepwise regression. Para isto, utilizou-se a função ***stepwise,*** do *Statistics and Machine Learning Toolbox* do Matlab, que executa o algoritmo e apresenta os resultados em uma interface amigável, permitindo a fácil interpretação dos resultados. A cada etapa, uma nova variável é adicionada ao modelo, sendo que o algoritmo testa as variáveis e seleciona aquela cujo impacto é o maior no sentido de diminuição do erro do modelo em relação à variável de resposta. Posteriormente, o algoritmo realiza uma seleção por remoção, analisando a possível exclusão de uma variável redundante do modelo. A seleção de variáveis foi realizada com os dados de treinamento previamente separados (80% do total).

3.2.5 *Desenvolvimento, validação e seleção do modelo*

Seguindo a metodologia proposta por KADLEC *et al.* (2009), em um primeiro momento procedeu-se o desenvolvimento de um modelo mais simples, utilizando para isso uma Regressão Linear Multivariável, seguindo pela elaboração de uma Rede Neural Artificial, gerando um modelo não-linear de mais alta complexidade. Destaca-se que o desenvolvimento de todos os modelos foi feito utilizando-se a mesma base de dados – aquela destinada previamente para este fim. A validação de todos os modelos foi realizada com os dados restantes, e a comparação foi feita analisando-se os valores dos coeficientes de determinação R^2 dos valores de resposta gerados pelo modelo em comparação com os valores reais da variável de resposta. Foram elaborados modelos com mais de um grupo de variáveis secundárias, a fim de analisar e validar os resultados gerados pelo algoritmo *step wise* previamente descrito.

As características de cada modelo, bem como a metodologia utilizada para os seus desenvolvimentos serão descrita na sequência.

3.2.5.1 Regressão Linear Multivariável (MLR)

GELADI & KOWALSKI (1986) descrevem o problema de uma regressão linear multivariável da seguinte forma: as medições realizadas no sistema geram um grupo de m variáveis independentes x_j $(j = 1,2, \dots, m)$ e uma variável dependente y, também chamada de variável de resposta. O objetivo é obter uma relação de primeira-ordem entre essas variáveis, o que pode ser expresso matematicamente da seguinte maneira:

$$y = b_0 + b_1 x_1 + b_2 x_2 + b_3 x_3 + \cdots + b_m x_m + e \tag{3.1}$$

onde b_0 é o termo independente, b_j são os coeficientes lineares das variáveis independentes x_j, e e é o resíduo, ou erro. A expressão também pode ser escrita na sua forma condensada:

$$y = b_0 + \sum_{j=1}^{m} b_j x_j + e \tag{3.2}$$

Vetorialmente, o problema se dá pela seguinte expressão:

$$y = x'b + e \tag{3.3}$$

onde y é um escalar, x' é o vetor transposto das variáveis independentes (sendo $x_0 = 1$), b é um vetor coluna (sendo b_0 o termo independente), e e é um escalar.

Entretanto, essas expressões descrevem a relação entre a variável de resposta e as variáveis independentes para somente uma amostra de dados, o que pode ser interpretado como um instante do tempo. Para um grupo de dados que contenha n amostras, ou instantes, o problema torna-se matricial e precisa-se relacionar as variáveis de resposta y_i $(i = 1,2, \dots, n)$, que compõem o vetor coluna y, com as mesmas m variáveis independentes $x_{i,j}$, que agora estarão representadas em uma matriz X composta por n linhas, referentes ao número de amostras, e m colunas, referentes ao número de variáveis independentes. O vetor coluna b engloba os coeficientes da regressão, e o vetor coluna e, os resíduos.

$$y = Xb + e \tag{3.4}$$

O vetor de resíduos, portanto, será expresso pela diferença entre as variáveis de resposta y e os resultados obtidos pelo produto entre as variáveis independentes X e os seus respectivos coeficientes de regressão b:

$$e = y - Xb \tag{3.5}$$

A Figura 3.3 facilita o entendimento do problema matricial:

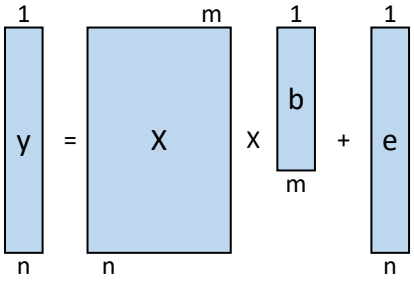

Figura 3.3: Representação gráfica de um problema MLR (Adaptado de GELADI & KOWALSKI, 1986).

GELADI & KOWALSKI (1986) ainda destacam três situações que podem ocorrer durante a aplicação do método MLR, dependendo do número de amostras (n) e do número de variáveis independentes (m):

- $m > n$: Existem mais variáveis independentes do que amostras. Neste caso, haverá um número infinito de soluções para o vetor b que se adequam à equação, o que não é desejado.

- $m = n$: O número de variáveis independentes e amostras é igual. Na prática, essa situação dificilmente será encontrada, mas ela fornece uma solução única para b, desde que a matriz X não seja posto-deficiente. A consequência disso é que o vetor de erro e é zerado, ou seja:

$$e = y - X.b = 0 \tag{3.6}$$

- $m < n$: Existem mais amostras do que variáveis independentes, o que faz com que exista mais de uma solução para b. A solução do problema é obtida ao

minimizar-se o vetor de erros e, o que normalmente é feito utilizando-se o Método de Mínimos Quadrados, cuja solução é obtida pela Equação 3.7.

$$b = (X'X)^{-1}X'y \tag{3.7}$$

Potencialmente, a inversa $(X'X)^{-1}$ pode não existir, o que dá origem ao problema conhecido como colinearidade, determinante zero ou de singularidade.

Para o presente trabalho, o problema localizou-se no caso em que $m < n$, dado o grande volume de amostras (n) frente ao baixo número de variáveis (m). O desenvolvimento do modelo foi realizado em Matlab utilizando o Método de Mínimos Quadrados.

Primeiramente, selecionaram-se as variáveis independentes da base de dados a serem utilizadas no modelo (**Xt**), baseado no estudo prévio da seleção de variáveis. Deve-se organizar os dados de modo a se obter uma matriz **U** com todas as variáveis independentes selecionadas (**Xt**), bem como uma coluna extra para ajuste dos termos independentes, não atrelados a nenhuma variável. Inseriu-se, portanto, uma coluna preenchida por valores unitários na primeira coluna da matriz **U** para esta finalidade. O vetor **yt** compreendeu as variáveis de resposta, e o vetor **b** os coeficientes de regressão. Com o operador "\", ajustaram-se os coeficientes de regressão e permitiu-se calcular os valores preditos pelo modelo com os dados de treinamento (**yCalc**). A comparação com os valores reais da variável de resposta foi feita pela geração de um gráfico comparando-as, e pela análise do valor de R^2 do mesmo.

Posteriormente, realizou-se a validação do modelo gerado, utilizando a base de dados destinados para este fim (**xv** e **yv**). Procedeu-se da mesma maneira que anteriormente, desta vez utilizando os coeficientes gerados previamente (**b**) para geração dos dados de resposta de validação (**yVal**).

Este procedimento foi realizado para três grupos de variáveis secundárias, a serem mostrados no Capítulo 4 (Resultados e Discussões).

3.2.5.2 Redes-neurais artificiais (ANN)

Diferentemente da técnica MLR, as Redes Neurais artificiais são modelos não-lineares de relativa complexidade, cuja estrutura tenta imitar a funcionalidade do cérebro humano. QIN (1997) destaca a grande utilidade de Redes Neurais para a modelagem de sistemas industriais e consequente aplicação para controle de processos, sendo amplamente aceitas na indústria de processos devido ao seu excelente custo-benefício, não-linearidade e o fato de serem desenvolvidas a partir de uma análise de dados.

WILLIS *et al.* (1992) afirmam que, embora inúmeras arquiteturas de Redes Neurais tenham sido propostas, a Rede Neural artificial do tipo *feed-forward* (FANN) é a mais utilizada para a modelagem de processos. HORNIK *et al.* (1989) e CYBENKO (1989) provaram que uma FANN é capaz de modelar qualquer função contínua de maneira satisfatória. Em virtude dos pontos expostos acima, este trabalho focou nas Redes Neurais FANN, já que o objetivo de sua aplicação é para a indústria de processos.

Uma FANN é composta por diversos neurônios, também chamados de nós, que são organizados nas camadas que fazem parte da Rede Neural. Os neurônios se comunicam para transferência de dados, de acordo com a maneira com que a FANN for treinada, e tornam-se capazes de contornar a não-linearidade do sistema através desta comunicação neuronal. A Figura 3.4 mostra a estrutura típica de uma rede do tipo FANN:

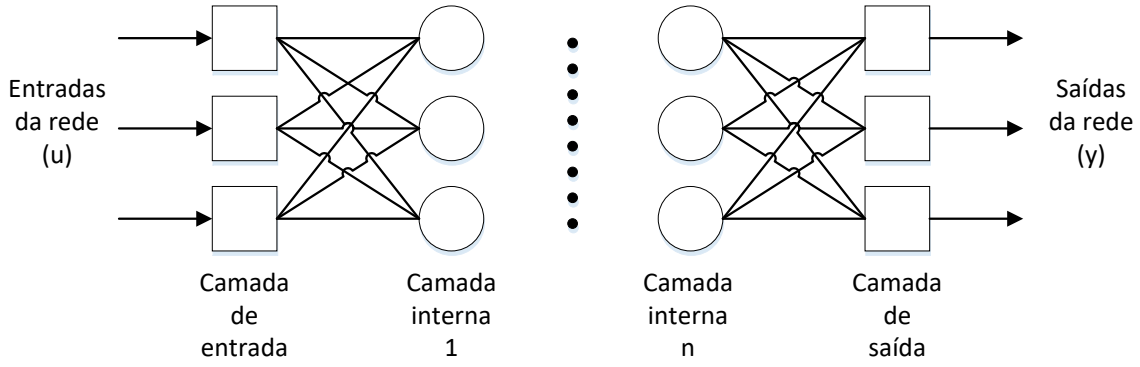

Figura 3.4: Estrutura de uma FANN (Adaptado de WILLIS *et al.*, 1992)

De maneira sucinta, o mapeamento realizado por uma Rede Neural do tipo FANN com duas camadas internas é demonstrado por WILLIS *et al.* (1992), da seguinte maneira: sejam $u(t)$ e $y(t)$, respectivamente, os dados de entrada e de saída do sistema. Por definição, a transformação funcional em cada camada da Rede Neural, J_i, pode ser representada pela combinação de uma transformação afim, dado pela soma ponderada das entradas para a camada posterior, W_i, com um mapeamento não-linear, geralmente utilizando-se a função sigmoidal, S. Sendo h_1 e h_2, respectivamente, os vetores de saída da primeira e segunda camadas internas da Rede Neural, y o vetor de saídas da rede e "." o operador de montagem, o equacionamento do problema é dado a seguir.

$$h_1 = J_1\big(u(t), y(t-\tau)\big) = S.W_1\big(u(t), y(t-\tau)\big) \tag{3.8}$$

$$h_2 = J_2\big(h_1(t)\big) = S.W_2\big(h_1(t)\big), \tag{3.9}$$

$$y(t) = J_0\big(h_2(t)\big) = S.W_0\big(h_2(t)\big) \tag{3.10}$$

Uma aproximação para o mapeamento não-linear desta FANN é dado por:

$$y(t) = J_0.J_2.J_1(u(t), y(t-\tau)) \tag{3.11}$$

O desenvolvimento de uma rede do tipo FANN começa com a definição da sua estrutura. Segundo a convenção definida por BREMMERMANN & ANDERSON (1989), uma FANN com quatro neurônios na camada de entrada, duas camadas internas com, respectivamente, sete e cinco neurônios, e um neurônio na camada de saída, irá ser identificada como uma rede (4-7-5-1).

Após definida a estrutura da FANN, a Rede Neural deve ser treinada com os dados de treinamento (previamente segmentados) de entrada e saída, com os dados sendo propagados pela rede de modo a ajustar os pesos internos de cada neurônio em cada camada. O procedimento é realizado objetivando-se minimizar o erro quadrático entre a saída obtida pela Rede Neural e o valor da variável de resposta original, sendo possível a

utilização de diversos algoritmos de otimização para tal tarefa. Com o modelo obtido, a Rede Neural é validada através da utilização dos dados de validação (WILLIS *et al.*, 1992).

O desenvolvimento das FANNs foi realizado em Matlab, utilizando-se dos recursos da interface **nntool**, disponível para elaboração e treinamento de Redes Neurais. A criação de uma **FANN** foi realizada utilizando-se a função **newff**, utilizando-se as variáveis de entrada pré-selecionadas (**Xt**) e as variáveis de resposta (**yt**), chamadas de variáveis-alvo no contexto de Redes Neurais. O treinamento da rede foi realizado de maneira iterativa até 50 épocas (iterações), utilizando para isso a função **train** com o algoritmo de otimização de Levenberg-Marquardt, cuja sequência iterativa realiza o ajuste dos parâmetros de cada neurônio. Após o treinamento da Rede Neural, procedeu-se a utilização do modelo não-linear com os dados de treinamento, obtendo-se as variáveis de resposta do modelo (**yCalc**) através da função **sim**. Analisou-se o desempenho da rede desenvolvida através do gráfico de **yCalc** e **yt** e da análise do valor de R^2 do mesmo.

As Redes Neurais foram desenvolvidas com apenas uma camada interna. Visto que as Redes Neurais apresentam um comportamento não-linear, é necessário que o seu treinamento se dê de forma adequada para que o seu desempenho seja satisfatório. Optou-se pela realização de repetições dos treinamentos, variando-se o número de iterações (**i**) em 20, 50 e 100 vezes na etapa de treinamento, posteriormente analisando-se o efeito deste parâmetro na convergência do treinamento das Redes Neurais. Na sequência, validou-se a Rede Neural através da utilização dos dados previamente destinados para este fim (**Xv** e **yv**). Para tal, utilizou-se a rede FANN com os dados de validação para geração de novas variáveis de respostas (**yVal**) e comparou-se com os dados **yv**. Este procedimento foi realizado com os mesmo três conjuntos de entrada utilizados nos modelos MLR, a fim de analisar-se o efeito destas variáveis na capacidade de predição com modelos não-lineares.

Com a escolha do grupo de variáveis e definição do número de iterações mais adequados para utilização na FANN, analisou-se o impacto do número de neurônios na camada interna da Rede Neural, variando o mesmo de 1 até 20 neurônios. Devido à inicialização randômica dos seus parâmetros, a Rede Neural pode não convergir sempre para o mesmo resultado durante o seu treinamento. Assim, o procedimento se deu pela inicialização de 100 Redes Neurais para cada número determinado de neurônios, com o treinamento das mesmas ocorrendo até 50 épocas. As redes de melhor desempenho, para cada número de neurônios, foram salvas e comparadas, permitindo a avaliação da influência da estrutura da FANN no seu desempenho.

4 Resultados e Discussões

Os dados coletados, descritos no Capítulo 3, foram organizados em uma matriz no software Matlab, com as variáveis independentes descritas nas Tabela 4.1, as quais serão subsequentemente mencionadas pela sua posição na matriz (X_i).

Tabela 4.1: Dados coletados e sua organização no Matlab

Descrição	Símbolo	Posição
Massa Específica Vaso de Diluição	ρ_{VD}	X_1
Temperatura Vaso de Diluição	T_{VD}	X_2
Nível Tanque Pulmão	L_{TP}	X_3
Temperatura Tratamento Térmico	T_{TT}	X_4
Temperatura Câmara Flash	T_{CF}	X_5
Pressão Câmara Flash	P_{CF}	X_6
Nível Câmara Flash	L_{CF}	X_7
Viscosidade Câmara Flash	μ_{CF}	X_8
Massa Específica Câmara Flash	ρ_{CF}	X_9

4.1 Pré tratamento de dados

Conforme descrito na metodologia, foi decidido pela eliminação de *outliers* e dados faltantes. Para as variáveis obtidas por sensores, vale destacar a eliminação de grupos de dados devido à identificação de *outliers* óbvios nos dados de massa específica: alguns pontos apresentavam valores menores do que a massa específica da água pura (1000 kg/m³), o que se mostra impossível considerando que o produto é composto por uma mistura de água e proteína de soja, cuja massa específica é, pelo histórico da planta e experiência da operação, sempre maior do que a da água pura. Além destes, os demais *outliers* encontrados foram do tipo não-óbvios, destacando-se pontos com valores de viscosidade muito baixa (na faixa de até 20 cP) que contrariavam o resto dos valores (a partir de 40 cP).

Para os dados de teor de sólidos, cujos valores são obtidos por análises laboratoriais, a identificação de *outliers* foi realizada de maneira qualitativa, não havendo uma métrica para decisão do que era uma variação de processo e o que era um possível desvio de resultado na análise laboratorial. Consultou-se a operação da planta para auxílio nesta identificação, levando em conta basicamente a experiência dos operadores acerca dos pontos destoantes.

Originalmente, haviam sido coletados 401 pontos de processo, já se descartando os pontos faltantes. Após o pré-tratamento de dados e análise de *outliers*, foram mantidos 371 pontos, representando uma eliminação de 7,5% dos pontos adquiridos. A partir dos dados restantes, foi realizada a segmentação de dados em dados para treinamento (80%) e validação (20%), correspondendo a, respectivamente, 302 e 69 pontos de processo.

4.2 Seleção de variáveis

O resultado da seleção de variáveis pela função **stepwise** no Matlab é apresentado em uma interface amigável, na qual o usuário pode acompanhar cada etapa do algoritmo, atentando para o efeito da variável acrescentada ao modelo. O erro do modelo vai sendo diminuído à medida que mais variáveis são acrescentadas, até chegar ao ponto em que o acréscimo de variáveis não melhora o desempenho do modelo.

Para o trabalho proposto, o modelo apresentou um aumento significativo no seu desempenho até a 6ª etapa do algoritmo, ponto no qual 5 variáveis constam no modelo (a 1ª etapa considera o modelo com apenas um coeficiente independente). As variáveis adicionadas ao modelo até este ponto, conforme indicado pelo ponto azul ao lado das mesmas na Figura 4.1, foram X_1, X_2, X_4, X_5 e X_8, e o algoritmo apresentou um valor de R^2 de 0,9244.

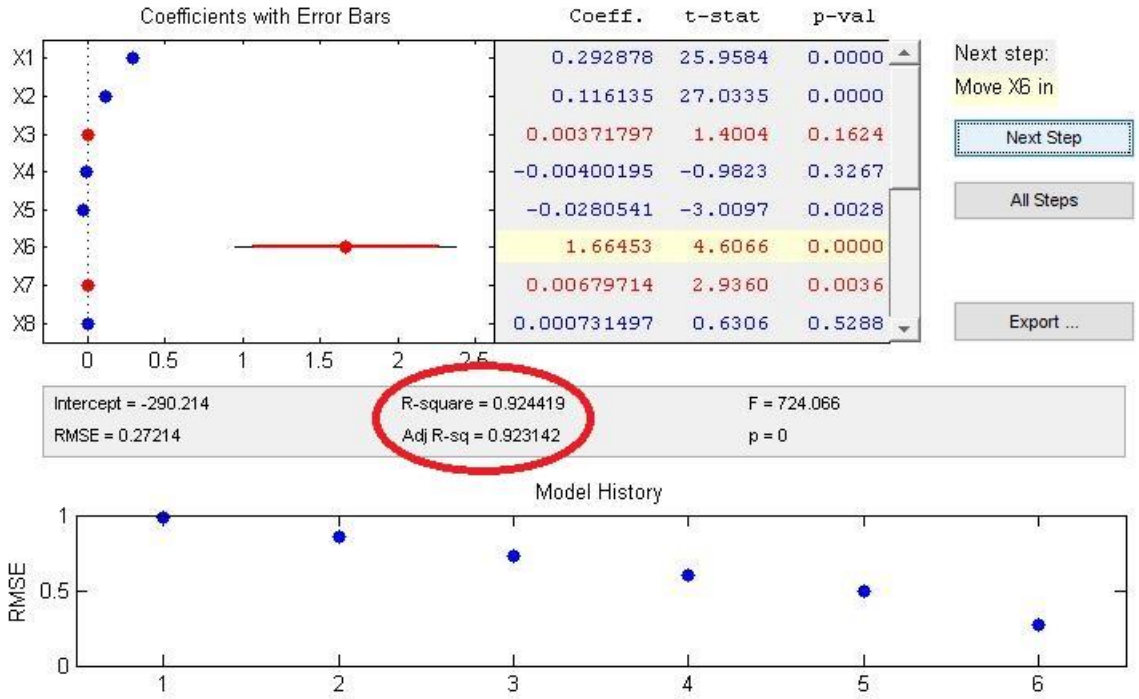

Figura 4.1: Resultado da seleção de variáveis até a 6ª etapa do algoritmo stepwise

A partir deste ponto, a adição de mais duas variáveis (X_6 e X_9), melhorou o resultado de R^2 até chegar a 0,9336. Por este motivo, decidiu-se elaborar modelos com os dois diferentes grupos de variáveis, a fim de analisar se o incremento destas variáveis extras poderia ter um impacto significativo nos modelos criados.

No início do trabalho, com uma base de dados diferente, à qual não se havia submetido um pré-tratamento de dados adequado, realizou-se o mesmo procedimento de seleção de variáveis pelo algoritmo **stepwise**. Neste momento, o resultado da análise indicou a

utilização das variáveis X_1, X_2, X_4, X_6, X_8 e X_9 para obtenção de um modelo linear de bom desempenho. Apesar de ter se decidido abandonar a base de dados por ela não se adequar ao requerido para o trabalho, resolveu-se pela análise do grupo de variáveis gerado nesta análise, visto que a única diferença em relação ao grupo de variáveis gerado pelo **stepwise** na base de dados utilizada é a ausência da variável X_5.

4.3 Modelo de Regressão Linear Multivariável (MLR)

A Tabela 4.2 apresenta os valores de R^2 na comparação entre os valores dos dados de treinamento calculados pelos modelos MLR desenvolvidos e os valores das variáveis de resposta medidos em laboratório, além de implementar a denominação daqui em diante usada para os grupos de variáveis.

Tabela 4.2: Valores de R^2 encontrados no desenvolvimento de modelos MLR.

Grupo	Variáveis	R^2
A	X_1, X_2, X_4, X_5 e X_8	0,9244
B	$X_1, X_2, X_4, X_5, X_6, X_8$ e X_9	0,9336
C	X_1, X_2, X_4, X_6, X_8 e X_9	0,9330

O aumento do número de variáveis aumentou sensivelmente o valor de R^2. Os resultados utilizando os modelos desenvolvidos com os dados de validação, no entanto, não seguiram este padrão, pois o aumento do número de variáveis não causou um maior valor de R^2 do modelo. A Figura 4.2 apresenta a comparação entre os resultados dos três modelos aplicados aos dados de validação, em relação aos resultados das medições de laboratório, ou seja, os valores que o modelo buscava atingir. A reta y=x serve como guia para a análise do gráfico, pois quanto mais próximo o ponto estiver da mesma, mais próximo o valor calculado (y) estará do valor de laboratório (x), e por consequência melhor é o desempenho do modelo.

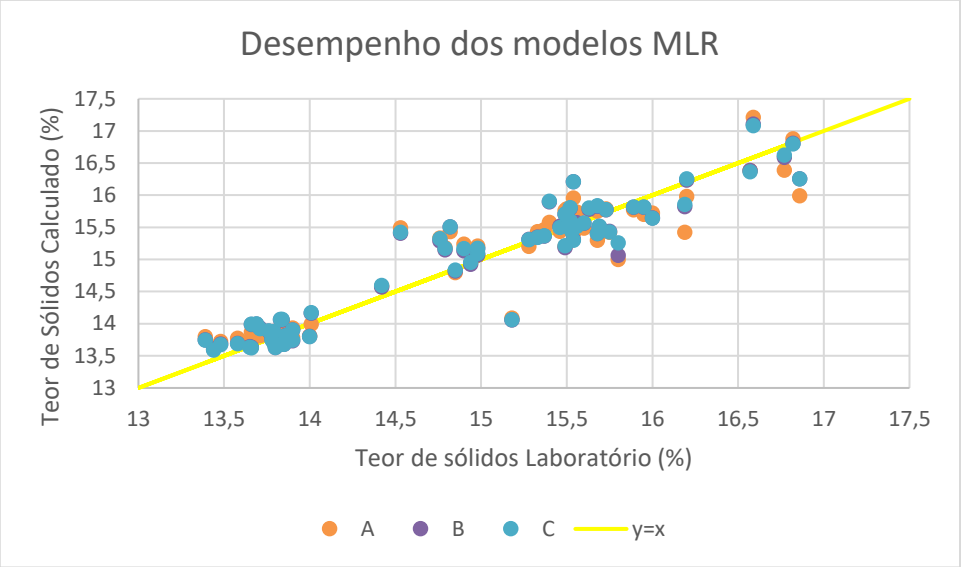

Figura 4.2: Validação dos modelos MLR.

A análise do gráfico permite verificar que os três modelos apresentaram resultados relativamente próximos àqueles das análises laboratoriais. Ao se compararem os valores de R^2, o grupo C apresenta o valor mais alto ($0{,}9045$) em comparação com $0{,}8871$ para o Grupo A e $0{,}9017$ para o Grupo B. Assim, pode-se dizer que o Grupo C de variáveis, composto por $\rho_{VD}, T_{VD}, T_{TT}, P_{CF}, \mu_{CF}$ e ρ_{CF} apresentou o melhor desempenho em um modelo MLR e seria o indicado para o desenvolvimento de um analisador virtual baseado neste tipo de modelagem. Este modelo é expresso pela Equação 4.1:

$$TS = -296{,}6263 + 0{,}2669\rho_{VD} + 0{,}1047T_{VD} - 0{,}0001\,T_{TT} + 1{,}6687P_{CF} - 0{,}0002\mu_{CF} + 0{,}0299\rho_{CF} \quad (4.1)$$

Ao calcular-se a diferença entre os valores preditos pelo modelo e os resultados de laboratório, chegou-se a um erro absoluto médio de 0,22 pontos percentuais de teor de sólidos, equivalente a um erro relativo de 1,46%. Sob o ponto de vista de aplicação na planta real, o modelo apresenta uma boa performance com os dados históricos, levando-se a concluir que a sua implementação tem grandes chances de resultar em um analisador virtual confiável para controle do processo.

Com o objetivo de compreender o modelo criado e as variáveis que o compõem, foi realizada uma análise do impacto que cada uma destas tem no resultado final de Teor de Sólidos gerado pelo modelo. Foram calculados os valores médios da base de dados para cada uma das variáveis, multiplicando-se estes valores pelo respectivo coeficiente do modelo: os resultados desta multiplicação correspondem à contribuição que cada variável tem no resultado calculado. Assim permite-se comparar a ordem de grandeza e a importância de cada variável no modelo desenvolvido. Os dados gerados estão apresentados na Tabela 4.3.

Tabela 4.3: Análise do impacto das variáveis no modelo MLR.

Variável	$\boldsymbol{\rho_{VD}}$ (kg/m³)	$\boldsymbol{T_{VD}}$ (°C)	$\boldsymbol{T_{TT}}$ (°C)	$\boldsymbol{P_{CF}}$ (bara)	$\boldsymbol{\mu_{CF}}$ (cP)	$\boldsymbol{\rho_{CF}}$ (kg/m³)
Valor médio	1037,8	32,2	140,0	0,6	54,3	1017,5
Coeficiente	0,2669	0,1047	-0,0001	1,6687	0,0002	0,0299

Impacto	276,99	3,37	-0,01	0,95	0,01	30,42

Percebe-se que o maior impacto no modelo para inferência do Teor de Sólidos do produto se dá pelas variáveis ρ_{VD} e ρ_{CF}, o que é bem coerente considerando-se que a massa específica é diretamente relacionada com a proporção entre solvente (água) e o produto sólido. Além disso, a ordem de grandeza do valor gerado pela variável ρ_{VD} (276,99) é muito próxima do termo independente do modelo (-296,62), o que torna evidente a sua grande influência no resultado da predição.

4.4 Modelo de Redes Neurais

Os modelos de Redes Neurais foram desenvolvidos com os Grupos de variáveis A, B e C, verificando-se também o efeito do número de iterações no processo de treinamento dos modelos. A Tabela 4.4 apresenta os valores de R^2 obtidos pela comparação entre os valores obtidos com os modelos desenvolvidos e os valores reais da variável de resposta, com a base de dados de treinamento.

Tabela 4.4: Desempenho das Redes Neurais treinadas.

Grupo	Número de iterações	R^2	R^2 médio
A	20	0,9422	0,9454
A	50	0,9515	
A	100	0,9434	
B	20	0,9465	0,9435
B	50	0,9494	
B	100	0,9346	
C	20	0,9463	0,9502
C	50	0,9556	
C	100	0,9494	

Os três grupos de variáveis apresentaram um bom desempenho no treinamento da Rede Neural, com valores de R^2, sempre maior que 0,94. Comparando-se os três grupos, o Grupo C apresentou um maior valor médio nos treinamento com diferentes números de iterações.

Em relação à variação no número de iterações, percebe-se que com 50 iterações os três grupos de variáveis apresentaram os valores de R^2 ligeiramente mais elevados, indicando que este é um bom número de iterações para o treinamento das Redes Neurais com estas variáveis. Assim, decidiu-se proceder para a etapa de análise do impacto do número de neurônios nos desempenhos das Redes Neurais com o grupo de variáveis C, que apresentou o melhor desempenho tanto nos modelos MLR quanto na etapa de treinamento das Redes Neurais. Ainda, definiu-se o número de iterações para treinamento das FANN em 50.

Para esta análise, variou-se o número de neurônios na única camada interna das Redes Neurais de 1 até 20, sendo que foram geradas e treinadas 100 Redes Neurais para cada caso, mantendo-se apenas a de melhor desempenho para a posterior comparação. O procedimento mostrou-se bastante eficiente, gerando modelos com elevados valores de R^2. A Figura 4.3 apresenta os resultados obtidos pelos modelos de melhor desempenho para cada número de neurônios na camada interna da Rede Neural.

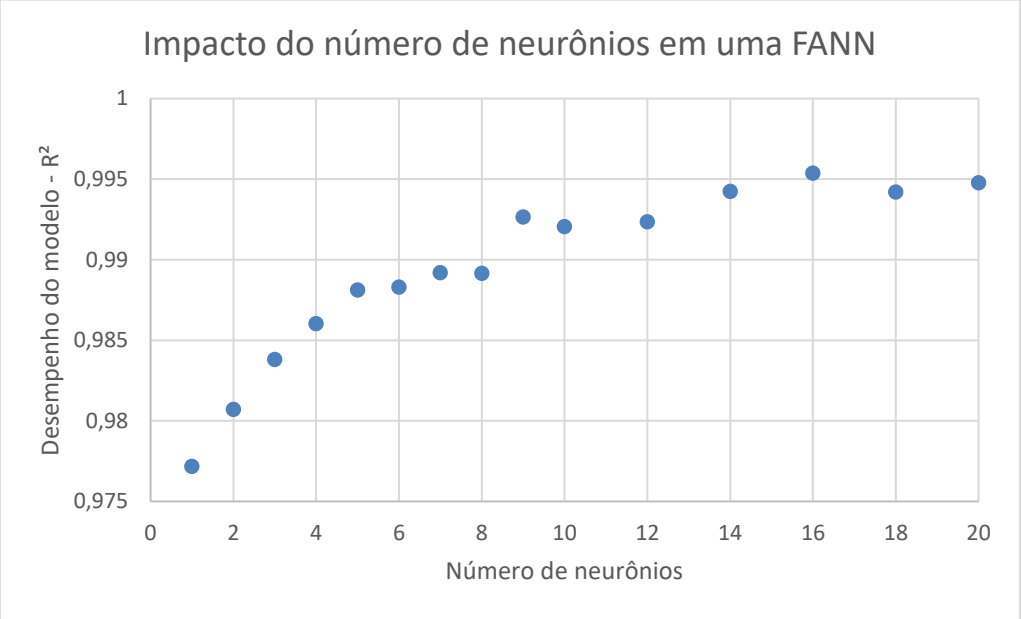

Figura 4.3: Análise do impacto do número de neurônios na camada interna de uma FANN.

Percebe-se um crescimento acentuado nos valores de R^2 no início do gráfico, para um número baixo de neurônios na camada interna. À medida que o número de neurônios já não é tão pequeno, o aumento no desempenho torna-se menos relevante, também em função de o valor de R^2 estar em um patamar bem elevado (acima de 0,99). Deste comportamento logarítmico, conclui-se que o aumento do número de neurônios na camada interna de uma Rede Neural tem, sim, um impacto positivo no sentido de melhorar o ajuste do modelo com os dados de treinamento, mesmo que a partir de certo ponto (9 neurônios, neste caso) a melhora já não seja mais significativa.

A validação dos modelos, assim como com os modelos MLR, foi realizada aplicando-se os dados de validação ao modelo e observando os valores de R^2 resultantes da comparação com os valores reais da variável de resposta. Este procedimento foi feito com todos os modelos, com os diferentes números de neurônios na camada interna, e os resultados podem ser observados na Figura 4.4.

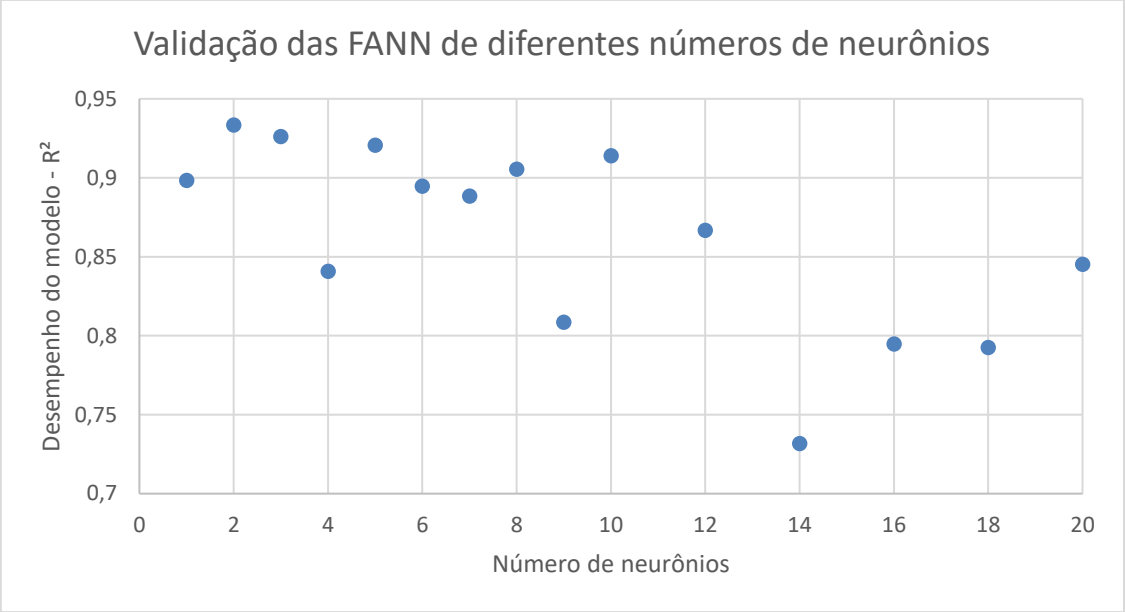

Figura 4.4: Validação dos modelos de FANN gerados com diferentes camadas internas.

Pode-se observar uma diminuição bastante significativa no desempenho dos modelos gerados, quando aplicados a uma base de dados diferente daquela com as quais a Rede Neural fora treinada. Além disso, os modelos com baixo número de neurônios, desta vez, apresentaram valores de R^2 mais altos do que os modelos com maior número de neurônios. A Tabela 4.5 apresenta a comparação entre os valores de R^2 obtidos no treinamento dos modelos e nas suas validações, bem como a diferença entre eles.

Tabela 4.5: Comparação das FANN desenvolvidas com diferentes camadas internas.

Núm. de neurônios	R² Treinamento	R² Validação	Diferença absoluta
1	0,9772	0,8983	0,0789
2	**0,9807**	**0,9334**	**0,0474**
3	**0,9838**	**0,9262**	**0,0576**
4	0,9860	0,8409	0,1451
5	**0,9881**	**0,9206**	**0,0675**
6	0,9883	0,8947	0,0936
7	0,9892	0,8883	0,1009
8	0,9892	0,9055	0,0836
9	0,9927	0,8087	0,1840
10	0,9921	0,9140	0,0781
12	0,9924	0,8667	0,1257
14	0,9943	0,7318	0,2624
16	0,9954	0,7948	0,2006
18	0,9942	0,7925	0,2017
20	0,9948	0,8453	0,1495

Os modelos com grande número de neurônios na camada interna apresentaram um desempenho bastante aquém do esperado, com uma grande diferença entre os valores de R^2 com os dados de treinamento e de validação. Este comportamento pode ser explicado

pelo efeito de "*overfitting*", no qual um modelo não-linear é ajustado com muita precisão e torna-se demasiadamente restrito às condições para as quais foi desenvolvido, não apresentando um bom desempenho com certos dados.

Considerando os resultados obtidos, decidiu-se pela utilização do modelo com dois, três ou cinco neurônios na camada interna da FANN, pois estes apresentam um bom desempenho tanto na etapa de treinamento (R^2 maior que 0,98) quanto na de validação (R^2 maior que 0,92). O erro médio absoluto obtido pelo modelo FANN com as variáveis do Grupo C e dois neurônios, quando se comparam os valores encontrados pelo modelo com os valores das análises laboratoriais de teor de sólidos foi de 0,17 pontos percentuais. Em termos relativos, este valor corresponde a um erro de 1,15%, na média. Para os modelos com três e cinco neurônios, os erros médios foram de, respectivamente, 0,18 e 0,19 pontos percentuais de teor de sólidos, correspondendo a um erro relativo de 1,21% e 1,29%.

De maneira geral, percebe-se que os modelos não-lineares de Redes Neurais apresentam uma excelente capacidade de adaptação aos dados propostos para treinamento e ajuste dos seus parâmetros. Todos os modelos desenvolvidos apresentaram um R^2 de no mínimo 0,9400 com os dados de treinamento, valor superior a qualquer modelo linear MLR desenvolvido previamente. Com o aumento do número de inicializações das Redes Neurais durante o treinamento, o desempenho foi ainda melhor, com os valores de R^2 não baixando de 0,97 para todos os diferentes números de neurônios testados. Entretanto, o desempenho destes modelos, quando aplicados a dados diferentes daqueles utilizados para treinamento, sofre uma queda significativa e pode os tornar similares aos modelos lineares de regressão multivariável.

Observando os resultados obtidos pelos melhores modelos na etapa de validação, o modelo MLR obteve um valor de R^2 de 0,9045, com um erro absoluto médio de 0,22 pontos percentuais de teor de sólidos, correspondendo a um erro relativo de 1,46%. Já o modelo não-linear de Redes Neurais de melhor desempenho gerou um coeficiente de determinação de 0,9334, com erro médio absoluto de 0,17% e erro relativo médio de 1,15%. No geral, ao se compararem os desempenhos dos modelos MLR e de Redes Neurais do tipo FANN, pode-se primeiramente concluir que os modelos FANN são superiores aos MLR, desde que treinados e escolhidos de maneira adequada. Por outro lado, os modelos MLR são mais simples, facilmente desenvolvidos e consequentemente de mais fácil manutenção e, na maioria das vezes, apresentam desempenhos também bastante satisfatórios.

A Tabela 4.6 apresenta alguns resultados obtidos por pesquisadores com diferentes métodos de modelagem.

Tabela 4.6: Resultados da literatura para diferentes técnicas de modelagem no desenvolvimento de analisadores virtuais.

Autores	Método	Aplicação	R^2 obtido
LIN *et al.* (2009)	Experimental - não especificado	Viscosidade de leite	0,77 - 0,93
CHEN *et al.* (2001)	MLR	Secagem de mirtilo	0,8
BRUWER *et al.* (2007)	PLS	Textura de snacks	>0,96
CHEN *et al.* (2001)	ANN	Secagem de mirtilo	>0,99

Percebe-se que os valores de R^2 encontrados para os modelos MLR (0,90) estão acima dos apresentados por CHEN et al. (2001) e tão bons quanto os de LIN et al. (2009), muito embora os autores não tenham especificado a técnica de regressão utilizada com os dados experimentais. Em comparação com o trabalho de BRUWER et al. (2007), que utiliza um modelo linear obtido pela técnica de regressão por Mínimos Quadrados Parciais (PLS), o modelo MLR aqui desenvolvido apresenta um desempenho inferior. MONTAGUE *et al.* (1992) desenvolveram um modelo MLR para estimar a concentração de biomassa em um biorreator contínuo, obtendo um resultado que permitiu a utilização do analisador virtual para controle do processo.

Em relação aos modelos FANN, o desempenho dos modelos desenvolvidos neste trabalho são similares aos encontrados na literatura, muito embora o trabalho de CHEN et al. (2001) tenha um resultado excepcional, com valores de R^2 muito próximos a 1. MONTAGUE *et al.* (1992) implementaram com sucesso um analisador virtual baseado em Redes Neurais para controle da produção de penicilina. Já o trabalho de LIU et al. (2010) compara a utilização de analisadores virtuais desenvolvidos por FANNs e por SVMs para estimação da concentração de biomassa em um fermentador e apresentou um resultado superior utilizando SVM, em comparação com FANN.

5 Conclusões e Trabalhos Futuros

O presente trabalho aplicou com sucesso uma metodologia para desenvolvimento de um analisador virtual para Teor de Sólidos, a ser aplicado em uma indústria produtora de proteína de soja isolada. Os resultados obtidos pelo algoritmo *stepwise regression* foram bastante proveitosos, tendo sido utilizados no desenvolvimento dos modelos. Os modelos de melhor desempenho, desenvolvidos a partir de Regressão Linear Multivariável (MLR) e Redes Neurais (não-linear), com diferentes grupos de variáveis, apresentaram erros relativos de 1,46% e 1,15%, respectivamente. Estes modelos podem ser implementados na indústria com chance de sucesso.

A efeito de comparação, os modelos de Redes Neurais apresentaram um desempenho superior aos de MLR. Entretanto, a maior simplicidade dos modelos MLR indicam que este pode ser o caminho mais indicado para utilização em uma planta real, considerando que os analisadores virtuais necessitam de uma manutenção contínua. Assim, o presente trabalho considera que o modelo MLR com as variáveis de entrada $\rho_{VD}, T_{VD}, T_{TT}, P_{CF}, \mu_{CF}$ e ρ_{CF} e o modelo FANN com dois neurônios em sua camada interna devem ser implementados na indústria como trabalho futuro, a fim de se monitorar o desempenho dos mesmos em um período longo e assim concluir qual o mais indicado para essa aplicação. Além disso, fica dentro deste escopo a manutenção dos analisadores virtuais, visto que este procedimento não pode ser desenvolvido neste trabalho por não ter sido possível implementar os analisadores virtuais na planta industrial.

Ainda, vale destacar que o presente trabalho possibilitou o aprendizado e implementação de uma metodologia razoavelmente simples para o desenvolvimento de analisadores virtuais, possibilitando que a mesma seja aplicada a diferentes outros problemas da Engenharia de Processos. Objetivamente, no futuro se deseja desenvolver um analisador virtual para estimação do Teor de Umidade do produto que sai do secador do tipo *Spray Dryer* descrito neste trabalho, reduzindo a necessidade de análises periódicas de tal parâmetro. Neste trabalho, deverá ser testado o desenvolvimento de modelos fenomenológicos com Filtros de Kalman e modelos não-lineares de Máquinas de Suporte Vetorial, a fim de analisar o possível desempenho superior dos mesmos na implementação em analisadores virtuais.

6 Referências

BANGA, J. R. *et al.* Improving food processing using modern optimization methods. **Trends in food science & technology**, 2003. v. 14, n. 4, p. 131–144. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0924224403000487>.

BREMMERMANN, H. J.; ANDERSON, R. W. An alternative to back-propagation: a simple rule for synaptic modification for neural net training and memory. **Department of mathematics, university of california, berkeley - internal report**, 1989.

BRUWER, M. J.; MACGREGOR, J. F.; BOURG, W. M. Soft sensor for snack food textural properties using on-line vibrational measurements. **Industrial and engineering chemistry research**, 2007. v. 46, n. 3, p. 864–870.

CHEN, C. R. *et al.* Drying technology : an prediction of quality changes during osmo-convective drying of blueberries using neural network models for. 2001. n. June 2012, p. 507–523.

CHTOUROU, M. *et al.* Control of a bioreactor using a neural network. **Bioprocess engineering**, 1993. n. 8, p. 251–254.

CYBENKO, G. Approximation by superpositions of a sigmoidal function. **Math. control signals systems**, 1989. n. 2, p. 303–314.

DENN, M. **Process modelling**. New York: Longman, 1986.

DOHNAL, M. Fuzzy bioengineering models. **Biotechnology and bioengineering**, 1985. v. 27, n. 8, p. 1146–1151.

EERIKÄINEN, T.; LINKO, P.; LINKO, S. Fuzzy logic and neural network applications in food science and technology. **… science & technology**, 1993. v. 41, n. August. Disponível em: <http://www.sciencedirect.com/science/article/pii/092422449390137Y>.

FACCHIN, S. U niversidade f ederal do r io g rande do s ul técnicas de análise multivariável aplicadas ao desenvolvimento de analisadores virtuais técnicas de análise multivariável aplicadas ao desenvolvimento de analisadores virtuais. 2005.

FILEV, D. P. *et al.* No title. **J. ferment. technol.**, 1985. n. 63, p. 545–553.

FORTUNA, L.; GRAZIANI, S.; XIBILIA, M. G. Soft sensors for product quality monitoring in debutanizer distillation columns. **Control engineering practice**, 2005. v. 13, n. 4, p. 499–508.

GABRYS, B. Learning hybrid neuro-fuzzy classifier models from data: to combine or not to combine? **Fuzzy sets and systems**, 2004. v. 147, n. 1, p. 39–56.

GELADI, P.; KOWALSKI, B. R. Partial least-squares regression - a tutorial. **Analytica chimica acta**, 1986. v. 185, p. 1–17. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/0003267086800289>.

GOVAERTS, R. *et al.* Control of an industrial spray drying unit. 1994. v. 2, n. 1, p. 69–85.

HAVLIK, S.; DEER, L.; OKOS, M. R. Computer aided engineering in the food industry. *In*: SINGH, R. P.; MEDINA, A. G. (Org.). **Food properties and computer-aided engineering of food processing systems**. Dordrecht: Springer Netherlands, 1989, p. 507–533. http://link.springer.com/chapter/10.1007%2F978-94-009-2370-6_39.

HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. **Neural networks**, 1989. v. 2, n. 5, p. 359–366.

KADLEC, P.; GABRYS, B.; STRANDT, S. Data-driven soft sensors in the process industry. 2009. v. 33, p. 795–814.

KUNCHEVA, L. I. **Combining pattern classifiers: methods and algorithms**. Hoboken, New Jersey: Wiley- IEEE, 2004.

LIN, K. **Soybeans: chemistry, technology, and utilization**. Singapore: Chapman & Hall, 1997.

LIN, T. I.; SOUZA, G. DE; YOUNG, B. **Towards a viscosity and density correlation for dairy fluids - a soft sensor approach**. [S.l.]: Elsevier Inc., 2009. V. 27.

LINKO, P.; ZHU, Y.-H. No title. **Kemia-kemi**, 1992. n. 19, p. 215–220.

LIU, G. *et al.* Model optimization of svm for a fermentation soft sensor. **Expert systems with applications**, 2010. v. 37, n. 4, p. 2708–2713. Disponível em: <http://dx.doi.org/10.1016/j.eswa.2009.08.008>.

MONTAGUE, G. A.; MORRIS, A. J.; THAM, M. T. Enhancing bioprocess operability with generic software sensors. **Journal of biotechnology**, 1992. v. 25, n. 1–2, p. 183–201.

NAKAMURA, T.; KURATANI, T.; MORITA, Y. No title. [S.l.]: [s.n.], 1985. p. 231–236.

OSORIO, D. *et al.* Soft-sensor for on-line estimation of ethanol concentrations in wine stills. **Journal of food engineering**, 2008. v. 87, n. 4, p. 571–577.

QIN, S. J. Chapter 8 neural networks for intelligent sensors and control | practical issues and some solutions. 1997. n. iv.

QUAD, I. I. M. S. A.; BOX, P. O. N. S. M. C. Dealing with missing data. 2002. p. 153–160.

SGARBIERI, V. C. **Proteínas em Alimentos Protéicos – Propriedades – Degradações – Modificações.** São Paulo: Livraria Varela, 1996.

SIMUTIS, R.; HAVLIK, I.; LIIBBERT, A. No title. **J. biotechnol.**, 1993. n. 27, p. 203–215.

THAM, M. T. *et al.* Soft-sensors for process estimation and inferential control. **Journal of process control**, jan. 1991. v. 1, n. 1, p. 3–14. Disponível em: <http://linkinghub.elsevier.com/retrieve/pii/095915249187002F>. Acesso em: 5 set. 2016.

TURUNEN, I. *et al.* Fuzzy modelling in biotechnology: sucrose inversion. **The chemical engineering journal**, 1985. v. 30, n. 3.

WILLIS, M. J. *et al.* Artificial neural networks in process estimation and control *. 1992. v. 28, n. 6, p. 1181–1187.