



UNIVERSIDADE D
COIMBRA

Tomás Miranda Pedrosa

Técnicas de Inteligência Computacional para Desenvolvimento de Sensores Virtuais em Processos Industriais

Dissertação no âmbito do Mestrado Integrado em Engenharia Eletrotécnica e de Computadores, no ramo de especialização de Automação, orientada pelo Prof. Dr. Rui Alexandre de Matos Araújo e pelo Dr. Jérôme Amaro Pires Mendes e apresentada ao Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Julho de 2022

1 2



9 0

UNIVERSIDADE DE
COIMBRA

Técnicas de Inteligência Computacional para Desenvolvimento de Sensores Virtuais em Processos Industriais

Tomás Miranda Pedrosa

Dissertação no âmbito do Mestrado Integrado em Engenharia Eletrotécnica e de Computadores, na especialização de Automação, orientada pelo Prof. Dr. Rui Alexandre de Matos Araújo e pelo Dr. Jérôme Mendes e apresentada ao Departamento de Engenharia Eletrotécnica e de Computadores da Faculdade de Ciências e Tecnologia da Universidade de Coimbra.

Orientadores: Prof. Doutor Rui Alexandre de Matos Araújo
Doutor Jérôme Amaro Pires Mendes

Júri

Presidente: Prof. Doutor Manuel Marques Crisóstomo
Vogais: Prof. Doutor Pedro Manuel Gens de Azevedo de Matos Faia
Prof. Doutor Rui Alexandre de Matos Araújo

Julho 2022

Agradecimentos

Gostaria de agradecer ao meu orientador, Professor Doutor Rui Alexandre de Matos Araújo, pela sua partilha de conhecimentos e contribuição neste projeto.

Ao meu co-orientador, Doutor Jérôme Amaro Pires Mendes, pelos seus conhecimentos, disponibilidade, dedicação e orientação ao longo deste trabalho.

À Celbi, na pessoa do Engenheiro Flávio Neto e à sua equipa, pela sua receptividade em participar e contribuir neste projeto, e aos Doutores António Jorge Pedrosa e Paulo Caetano que proporcionaram e coordenaram o protocolo com a FCTUC.

Ao projecto SmartWater “Smart Water Meter and Intelligent Water Analytics Platform for Water Pipeline Leakages Localization and End-Use Water Demand Outline” (SmartWater/2019/47730), co-financiado pelo “Quadro de Referência Estratégico Nacional Portugal 2020” (PT2020) pelo suporte ao desenvolvimento ao trabalho elaborado nesta dissertação.

Ao projeto iProMo Sistemas Inteligente para Controlo de Processos de Moagem (referência CENTRO-01-0247-FEDER-069730) pelo suporte ao desenvolvimento ao trabalho elaborado nesta dissertação.

À minha família e amigos por todo o apoio, motivação e força que me prestaram ao longo dos meus anos de estudo.

A todos,
Muito Obrigado!

Abstract

Nowadays, with the presence of Industry 4.0 and Big Data, one of the great interests of companies, scientists and others, is to explore the large volume of data that is acquired and stored by the industry. The industrial processes include several physical sensors whose function is to measure process variables (temperatures, pressures, conductivities, opening or closing of valves, etc.) in order to monitor and control the various stages of the process, however some variables cannot be determined without resorting to laboratory analysis. This way, costs are incurred for companies, the measures are sometimes impracticable, and information on the state of the system can be lost in real time, which consequently can lead to the degradation of the final product.

Here comes a solution that is increasingly gaining space in the industrial world and that, through its results, demonstrates that it is the present and the future, when it comes to predicting variables. We are talking about the soft sensor, a tool that is able to estimate process quality variables (output variable - target) through a model learned based on historical data. This data is normally provided or accessed from the process or laboratory database. When working with industrial data, we must bear in mind that the data is not always ready to be used directly in the predictive model, there is a whole set of tasks to implement, such as creating a database that stores the company's necessary data, the cleaning of the data itself (missing or faulty data, outliers, noise), input variable selection and model selection.

There are several predictive models in the literature, in order to deal with this diversity. We focused on three of them: variable selection methods, penalized regression methods and support vector machines. To be able to study these mentioned regression models and build a comparative procedure between them, a comparative methodology was developed in which all models are tested with the same set of data so that they start from the same level of equality. This methodology is a cross-validation that estimates the parameters and hyper-parameters of each model and evaluates the predictive capacity of each one.

In general terms, the analysis performed with datasets available in public repositories demonstrate that the support vector regression methods presented the best results, however the performance obtained by most of the regression models studied was interesting. Thus, we can conclude that they can in fact be important tools in supporting operational processes, making

the path more practical and the final product more efficient.

Keywords

Soft Sensors; Predictive Models; Computacional Intelligence; Prediction; Linear Regression and Non-Linear Regression.

Resumo

Nos dias de hoje, com a presença da Indústria 4.0 e de Big Data, um dos grandes interesses das empresas, dos cientistas de dados e não só, passa por explorar o grande volume de dados que são adquiridos e armazenados pela indústria. Os processos industriais incluem vários sensores físicos que têm como função registrar as variáveis processuais (temperaturas, pressões, condutividades, abertura ou fecho de válvulas, etc.) para se poder monitorizar e controlar as várias fases do processo: no entanto, algumas variáveis não se conseguem determinar sem se recorrer a análises laboratoriais. Por esta via, acarretam-se custos às empresas, as medidas por vezes são impraticáveis, podendo ocorrer perda de informação do estado do sistema em tempo real, o que conseqüentemente pode levar à degradação do produto final.

Aqui entra uma solução que cada vez mais está a ganhar o seu espaço no mundo industrial e que demonstra, pelos seus resultados, ser o presente e o futuro no que toca à previsão de variáveis. Falamos do sensor virtual, uma ferramenta que é desenvolvida com software capaz de estimar a variável de qualidade do processo (variável de saída - target) através de um modelo aprendido com base em dados históricos. Estes dados são normalmente acedidos à base de dados do processo ou do laboratório. Quando trabalhamos com dados industriais temos de ter em conta que nem sempre os dados estão prontos a serem utilizados diretamente no modelo preditivo, há todo um conjunto de tarefas para implementar, como a criação de uma base de dados que armazene os dados necessários da empresa, a própria limpeza dos dados (dados em falta ou falha, outliers, ruído, etc.), a seleção de variáveis de entrada e a seleção do modelo preditivo.

Existem diversas categorias de métodos preditivos na literatura, de forma a lidar com essa diversidade focou-se em três delas: métodos de seleção de variáveis, métodos de regularização e métodos de regressão por vetores de suporte. Para que fosse possível estudar estas categorias referidas e construir um procedimento comparativo entre elas, desenvolveu-se uma metodologia comparativa na qual todos os métodos são testados com o mesmo conjunto de dados para que partissem do mesmo pé de igualdade. Esta metodologia é uma validação-cruzada que estima os hiper-parâmetros de cada modelo de regressão e avalia a capacidade preditiva de cada um.

Em termos gerais, a análise feita com os conjuntos de dados disponíveis em repositórios públicos demonstram que os modelos de regressão por vetores de suporte apresentaram os me-

lhores resultados, no entanto o desempenho obtido pela maioria dos modelos estudados foi interessante. Assim, podemos concluir que podem ser de facto ferramentas importantes no apoio a processos operacionais, tornando o caminho mais prático e o produto final mais eficiente.

Palavras Chave

Sensores Virtuais; Modelos Preditivos; Inteligência Computacional; Previsão; Regressão Linear e Regressão Não Linear.

Conteúdo

1	Introdução	1
1.1	Motivação e contexto	2
1.2	Objetivos e Contribuições	4
1.3	Estrutura do documento	5
2	Visão Geral dos Sensores Virtuais	7
2.1	Sensores Virtuais em Processos Industriais	8
2.1.1	Tipo de Aplicações de Sensores Virtuais	8
2.2	Fases de Desenvolvimento de um Sensor Virtual	9
2.2.1	Recolha e Análise de Dados	9
2.2.2	Pré Processamento de Dados	10
2.2.3	Seleção de Variáveis	13
2.2.4	Seleção e Treino do Modelo	15
2.2.5	Validação do Modelo	17
2.2.6	Manutenção do Sensor Virtual	19
3	Modelos de Regressão Preditivos Aplicados aos Sensores Virtuais	21
3.1	Introdução	22
3.1.1	Problema de Regressão	22
3.2	Multiple Linear Regression	23
3.3	Métodos de Seleção de Variáveis	24
3.3.1	<i>Stepwise Regression</i>	24
3.4	Métodos de Regressão de Penalização	26
3.4.1	<i>Ridge Regression</i>	27
3.4.2	<i>Least Absolute Shrinkage and Selection Operator</i>	27
3.4.3	<i>Elastic Net</i>	28

Conteúdo

3.4.4	<i>Support Vector Regression</i>	28
3.5	Métricas de Desempenho Adotadas na Tese	31
3.5.1	Root Mean Square Error	31
3.5.2	Coefficiente de Determinação - R^2	31
3.6	Comparação dos Modelos	32
4	Resultados e Discussão	35
4.1	Introdução	36
4.2	Conjunto de Dados para Modelação do Sensor Virtual	36
4.3	Hiper-parâmetros dos Modelos de Regressão	37
4.4	Resultados	37
4.4.1	Resultados 'Box-Jenkins'	38
4.4.2	Resultados 'Friedman'	41
4.4.3	Resultados 'Boston Housing'	43
4.4.4	Resultados 'Debutanizer Column'	46
4.4.5	Resultados 'Cement Industry'	48
4.4.6	Resultados 'pH'	49
5	Conclusões	51
5.1	Conclusão	52
5.2	Trabalho Futuro	52
A	Anexo A - Gráficos Modelos de Regressão	59
A.1	Box-Jenkins	60
A.2	Friedman	62
A.3	Boston Housing	64
A.4	Debutanizer Column	66
A.5	Cement Industry	69
A.6	pH	71

Lista de Figuras

1.1	Desempenho preditivo do modelo vs Quantidade de dados [4].	3
2.1	Fases de Desenvolvimento de um SS [6]	9
2.2	Tipos de métodos de seleção de variáveis	13
2.3	Abordagens de Modelagem do SS	15
2.4	<i>5-fold cross-validation</i> [47]	18
2.5	<i>Leave-one-out Cross Validation</i>	18
3.1	Modelos de Regressão aplicados aos SSs	22
3.2	Procedimento do Stepwise Regression	25
3.3	Complexidade do modelo vs Erro de previsão [52]	26
3.4	Representação do modelo SVR	29
4.1	Resultados de treino e teste do SVR (Linear), com vista normal e vista ampliada (Box-Jenkins).	40
4.2	Resultados de treino e teste do SVR (rbf), com vista normal e vista ampliada (Box-Jenkins).	40
4.3	Resultados de treino e teste do RR, com vista normal e vista ampliada (Box-Jenkins).	41
4.4	Resultados de treino e teste do SVR (rbf), com vista normal e vista ampliada (Friedman).	42
4.5	Resultados de treino e teste do MLR, com vista normal e vista ampliada (Friedman).	43
4.6	Resultados de treino e teste do SR, com vista normal e vista ampliada (Friedman).	43
4.7	Resultados de treino e teste do SVR (rbf), com vista normal e vista ampliada (Boston Housing).	45
4.8	Resultados de treino e teste do SR, com vista normal e vista ampliada (Boston Housing).	45

Lista de Figuras

4.9	Resultados de treino e teste do MLR, com vista normal e vista ampliada (Boston Housing).	46
4.10	Resultados de treino e teste do SVR (rbf), com vista normal e vista ampliada (Debutanizer Column).	47
4.11	Resultados de treino e teste do SVR(rbf), com vista normal e vista ampliada (Cement Industry).	49
4.12	Resultados de treino e teste do LASSO, com vista normal e vista ampliada (Cement Industry).	49
4.13	Resultados de treino e teste do SVR(rbf), com vista normal e vista ampliada (pH).	50
A.1	Resultados de treino e teste do MLR (Box-Jenkins).	60
A.2	Resultados de treino e teste do SR (Box-Jenkins).	60
A.3	Resultados de treino e teste do LASSO (Box-Jenkins).	61
A.4	Resultados de treino e teste do EN (Box-Jenkins).	61
A.5	Resultados de treino e teste do RR (Friedman).	62
A.6	Resultados de treino e teste do LASSO (Friedman).	62
A.7	Resultados de treino e teste do EN (Friedman).	63
A.8	Resultados de treino e teste do SVR(Linear) (Friedman).	63
A.9	Resultados de treino e teste do RR (Boston Housing).	64
A.10	Resultados de treino e teste do LASSO (Boston Housing).	64
A.11	Resultados de treino e teste do EN (Boston Housing).	65
A.12	Resultados de treino e teste do SVR(Linear) (Boston Housing).	65
A.13	Resultados de treino e teste do MLR (Debutanizer Column).	66
A.15	Resultados de treino e teste do LASSO (Debutanizer Column).	66
A.14	Resultados de treino e teste do SR (Debutanizer Column).	67
A.16	Resultados de treino e teste do LASSO (Debutanizer Column).	67
A.17	Resultados de treino e teste do EN (Debutanizer Column).	68
A.18	Resultados de treino e teste do SVR(Linear) (Debutanizer Column).	68
A.19	Resultados de treino e teste do MLR (Cement Industry).	69
A.21	Resultados de treino e teste do LASSO (Cement Industry).	69
A.20	Resultados de treino e teste do SR (Cement Industry).	70
A.22	Resultados de treino e teste do EN (Cement Industry).	70
A.23	Resultados de treino e teste do SVR(Linear) (Cement Industry).	71
A.24	Resultados de treino e teste do MLR (pH).	71

A.25 Resultados de treino e teste do SR (pH).	72
A.26 Resultados de treino e teste do RR (pH).	72
A.27 Resultados de treino e teste do EN (pH).	73
A.28 Resultados de treino e teste do SVR(Linear) (ph).	73

Lista de Tabelas

3.1	Funções <i>Kernel</i> e o seu campo de aplicação	30
4.1	Seleção dos Hiper-parâmetros usados em cada um dos modelos de regressão durante a fase de treinamento do modelo.	38
4.2	Visão geral do Conjunto de Dados utilizados.	38
4.3	Número de amostras de CO_2 e parâmetros estatísticos.	38
4.4	Média das métricas de desempenho no Box-Jenkins.	39
4.5	Número de amostras de Output e parâmetros estatísticos.	41
4.6	Média das métricas de desempenho no Friedman.	41
4.7	Número de amostras de valores médios de casas em Boston e parâmetros estatísticos.	44
4.8	Média das métricas de desempenho no Boston Housing.	44
4.9	Número de amostras de Concentração de Butano e parâmetros estatísticos.	46
4.10	Média das métricas de desempenho no Debutanizer.	47
4.11	Número de amostras de 'Coarse Return' e parâmetros estatísticos.	48
4.12	Média das métricas de desempenho para os diferentes níveis de atrasos temporais	48
4.13	Número de amostras de pH e parâmetros estatísticos.	50
4.14	Média das métricas de desempenho no pH.	50

List of Algorithms

1	Pseudo-código da comparação preditiva dos modelos de regressão	33
---	--	----

Abreviaturas e Símbolos

ANN Artificial Neural Network

CQO Carência Química de Oxigênio

DL Deep Learning

EN Elastic Net

EM Expectation Maximization

GA Genetic Algorithm

HI Hampel Identifier

HI-SW Hampel Identifier with Sliding Window

IEEE Institute of Electrical and Electronics Engineers

LASSO Least Absolute Shrinkage and Selection Operator

LS Least Squares

MAD Median Absolute Deviation

MAE Mean Absolute Error

ML Machine Learning

MLP MultiLayer Perceptron

MLR Multiple Linear Regression

MSE Mean Squared Error

NFS Neuro-Fuzzy System

NN Neural Network

List of Acronyms

PCA Principal Component Analysis

PLS Partial Least Squares

RBF Radial basis function

RMSE Root Mean Squared Error

RPLS Recursive Partial Least Squares

RR Ridge Regression

SA Simulated Annealing

SR Stepwise Regression

SS Soft Sensor

SVM Support Vector Machine

SVR Support Vector Regression

SW Sliding Window

1

Introdução

Conteúdo

1.1	Motivação e contexto	2
1.2	Objetivos e Contribuições	4
1.3	Estrutura do documento	5

Este capítulo apresenta a motivação e contexto, objetivos e as principais contribuições da presente dissertação.

1.1 Motivação e contexto

De forma geral, os processos industriais encontram-se equipados com uma variedade de sensores, projetados para, de forma online, fazer supervisão, monitorização e controlo, medindo uma grande quantidade de variáveis (e.g. temperatura, fluxo, aberturas de válvulas) disponíveis a uma frequência constante com o objetivo final de obter um produto com a qualidade desejada. Algumas variáveis são críticas para o funcionamento do processo (variáveis de qualidade) mas nem sempre são medidas com uma frequência alta devido à falta ou elevado preço de obtenção dos sensores, levando à perda de informação do estado do sistema. Outras vezes, as variáveis são mensuráveis por análises em laboratório, com uma frequência de aquisição muito baixa, envolvendo atrasos entre a colheita de amostra e a obtenção de resultado de medição por laboratório. Para ultrapassar este problema, por vezes, a obtenção dessas variáveis é feita com recurso a análises laboratoriais, no entanto a frequência de medição irregular e os atrasos temporais substanciais, podem levar à degradação da qualidade do produto final [1].

Para responder a estas limitações, os Sensores Virtuais ou *Soft Sensor* (SS), têm ganho popularidade em ambiente industrial. Estes sensores são modelos inferenciais baseados em inteligência computacional, que fazem uso de variáveis do processo (disponíveis através de sensores), conhecidas como variáveis “fáceis de medir” (variáveis obtidas por sensores presentes no processo com alta frequência de aquisição, ex. 1ms a 5s), para estimar variáveis de qualidade “difíceis de medir” (ex. variáveis obtidas por laboratório com baixa frequência de aquisição, ex. 1 a 24 horas) [1]. Têm sido ferramentas importantes na estimação das variáveis críticas do processo com uma maior frequência, na deteção de falhas de sensores, nos sistemas de medição *back-up*, na análise *what-if*, levando a um eventual aumento significativo da qualidade do produto final e à redução dos custos envolvidos na sua criação [2].

Os SSs fornecem um conjunto de vantagens para as aplicações industriais, entre as quais destacam-se: (I) são uma alternativa de baixo custo comparada aos sensores de hardware que as indústrias contêm; (II) operam em paralelo com esses sensores, o que é uma mais valia para o processo de monitorização e controlo da indústria; (III) permitem estimar variáveis de qualidade em tempo real, aumentando a qualidade do processo e, conseqüentemente, o produto final.

Estes sensores, como foi referido anteriormente, têm como base inteligência computacional ou, também conhecida, aprendizagem de máquina (*Machine Learning* - ML). Cada vez mais a ML é popularizada no ambiente industrial muito graças à disponibilização dos dados e aos bons resultados que têm apresentado ao longo dos últimos tempos. *Deep learning* é um tipo de ML, que pode ser referido a arquiteturas de redes neuronais convencionais que são extremamente complexas. Os seus algoritmos simulam a capacidade de aprendizagem de um cérebro

humano. Um conjunto de dados é submetido a diversas camadas de processamento não lineares (*layers*), na qual a rede neuronal artificial aprende, com elevados níveis de precisão, conforme cheguem novas informações. Um dos principais fatores de sucesso da DL está relacionado com a quantidade de dados que se tem disponível para executar a sua aprendizagem, já que se obtém melhores resultados quanto maior for o conjunto de dados [3]. Como se pode ver na Figura 1.1, a quantidade de dados disponíveis é um parâmetro que faz variar o desempenho de um modelo, seja ele de ML ou DL. Repare-se que entre os três tipos de algoritmos apresentados, os modelos tradicionais de ML são os que melhoram de imediato o seu desempenho preditivo quando os dados são adicionados. No entanto, à medida que o conjunto de dados de treino vai sendo maior, a curva de aprendizagem vai ficando estagnada, concluindo que, para grandes quantidades de dados o modelo não tem como melhorar mais o seu desempenho. Para a mesma tarefa de aprendizagem, mas usando uma rede neuronal simples, ou seja, com poucas camadas de processamento, atinge-se um melhor desempenho de previsão. Utilizando uma rede neuronal complexa será perceptível que terá um maior desempenho preditivo do que as outras duas abordagens, após uma quantidade considerada de dados de treino. Resumindo, os modelos de *deep learning* tendem a aumentar o seu desempenho quando se aumenta a quantidade de dados de treino.

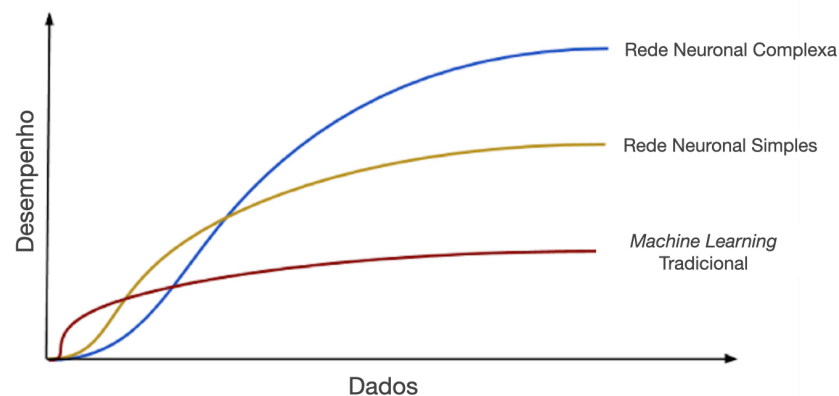


Figura 1.1: Desempenho preditivo do modelo vs Quantidade de dados [4].

Tendo em conta que o conjunto de dados, no contexto dos SS's, é frequentemente reduzido (desde de amostras de 2 em 2 horas a uma amostra por dia), as técnicas de DL são pouco atrativas e o foco ainda se mantém nas técnicas de regressão tradicionais de ML.

1.2 Objetivos e Contribuições

Esta dissertação tem como principal objetivo implementar e comparar técnicas de ML no contexto de modelagem de um sensor virtual. Os objetivos específicos são:

- Pesquisa e conhecimento prévio das técnicas de regressão;
- Implementar, testar e comparar algoritmos de aprendizagem supervisionada capazes de realizar regressão sobre um conjunto de dados de treino de tamanho reduzido:
 - A abordagem clássica, regressão linear múltipla (MLR);
 - *Stepwise Regression* (SR), inserida na categoria de seleção de variáveis;
 - Métodos de regressão penalizados que se inclui:
 - Ridge Regression* (RR);
 - Least Absolute Shrinkage and Selection Operator* (LASSO);
 - Elastic Net* (EN);
 - Support Vector Regression* (SVR).
- Desenvolver um ambiente comparativo para selecionar os parâmetros e hiper-parâmetros e avaliar o desempenho de cada um dos modelos de regressão em estudo;
- Aplicar os modelos implementados em vários conjunto de dados.

A dissertação foi desenvolvida no ambiente de programação Python, no qual incluímos algumas *toolboxes*, por exemplo, *pandas*, *numpy*, *scikit-learn*, entre outras.

Nesta dissertação é criado um ambiente de análise de dados industriais que inclui 4 estágios: aquisição de dados, tratamento de dados, pré-processamento de dados e modelação de dados. As 3 primeiras fases são de facto muito importantes já que são responsáveis por gerar dados adequados que serão usados na modelação, parte final e fulcral do processo. Foram estudados e testados, de maneira a se analisar e validar a sua relevância, modelos de regressão de aprendizagem supervisionada num conjunto de dados em contexto de sensores virtuais. Por fim, desenvolveu-se um ambiente de comparação o qual tem como objetivo analisar e comparar modelos de diferentes categorias preditivas (modelo tradicional de regressão, modelo de seleção de variáveis e modelos de regressão de penalização).

1.3 Estrutura do documento

Depois do capítulo introdutório, a revisão do estado da arte envolve questões como, aplicações industriais, modelos de regressão e metodologias para o desenvolvimento de um sensor virtual, pode ser encontrado no **Capítulo 2**.

O **Capítulo 3** apresenta todos os modelos de regressão usados em cada um dos conjuntos de dados, bem como a sua implementação. É ainda descrito o ambiente de computação para estimação dos hiper-parâmetros e comparação dos diferentes modelos.

O **Capítulo 4** descreve os conjuntos de dados usados para testar os modelos de regressão. São apresentados e discutidos os resultados obtidos para cada um dos modelos, em cada um dos conjuntos de dados;

Para finalizar, no **Capítulo 5** são apresentadas as conclusões acerca do trabalho desenvolvido, assim como possíveis trabalhos futuros.

Este documento inclui também 1 anexo nas páginas finais:

- Anexo A - Gráficos dos resultados dos modelos de regressão.

2

Visão Geral dos Sensores Virtuais

Conteúdo

2.1	Sensores Virtuais em Processos Industriais	8
2.2	Fases de Desenvolvimento de um Sensor Virtual	9

2. Visão Geral dos Sensores Virtuais

Neste capítulo apresenta-se a visão geral de sensores virtuais. É apresentada uma análise de todas as fases de desenvolvimento de um sensor virtual, identificam-se vários métodos computacionais utilizados em cada uma delas, e exemplos práticos, presentes no estado da arte.

2.1 Sensores Virtuais em Processos Industriais

Num ambiente industrial um dos maiores objetivos passa por procurar melhores soluções para que o processo industrial apresente os melhores resultados possíveis. O processo industrial está equipado com inúmeros sensores com o intuito de transmitir informação para a secção de monitorização e controlo do processo. Estes sensores físicos fazem a leitura de variáveis processuais (temperatura, pressão, pH, condutividade, etc) com frequências elevadas. Uma das limitações da monitorização do processo industrial surge quando se pretende fazer uma leitura automática de variáveis críticas do processo. Estas variáveis são relevantes e importantes, mas são medidas por vezes em laboratório, havendo atrasos significativos e possível perda de informação do estado do sistema.

Com os avanços da tecnologia e localizando-nos na era da Indústria 4.0 [5], os processos industriais são cada vez mais complexos e contêm maiores quantidades de dados para lidar. Tendo em conta este enquadramento, ao longo das últimas décadas vários investigadores têm desenvolvido modelos que se baseiam nos dados históricos do processo. Estes modelos são conhecidos como modelos inferenciais ou sensores virtuais, capazes de prever as variáveis críticas (ex. variáveis laboratoriais) do processo a partir das variáveis processuais (ex. sensores, atuadores, entre outros). A presença destes sensores nas indústrias petroquímicas e químicas, farmacêuticas, aço, papel e celulose é cada vez mais acentuada [6–10]. Por exemplo, na indústria da pasta de papel, existe a variável carência química de oxigénio (CQO) que é crítica ao processo, já que tem influência direta na qualidade da pasta obtida: esta é medida por análises laboratoriais, em alguns casos, três vezes por semana. Esta medida leva a atrasos temporais e pode resultar numa latência incomportável para sistemas de controlo automático.

Embora os autores dos artigos [6] e [7] possam não ir ao encontro da mesma filosofia, ambos os autores partilham a mesma comunidade no que toca às fases de desenvolvimento de um sensor virtual. Começam com a recolha de dados e aprendizagem do processo, de seguida analisam os dados recolhidos, partem para a estimação do modelo e, por último, validam o modelo obtido.

2.1.1 Tipo de Aplicações de Sensores Virtuais

Os sensores virtuais têm uma vasta gama de aplicações, o que permite que sejam utilizados em diferentes áreas industriais (por exemplo na indústria química e petroquímica, pasta e papel, refinarias, etc). A sua implementação pode resolver problemas de [11]:

- **Predição de variáveis em tempo real que não são medidas por sensores físicos** - na

maioria dos processos industriais existem variáveis de qualidade que são essenciais para garantir a qualidade da produção. Cenários com limitações técnicas ou económicas, inviabilizam a medição *online* dessas variáveis;

- **Deteção e diagnóstico de falhas no processo** - a equipa de supervisão é responsável por monitorizar o processo ao longo do tempo. A deteção e diagnóstico de falhas são algumas das suas funções. Com a incorporação de um SS no sistema, pode-se monitorizar o estado do processo, e em caso de desvio das condições normais, pode-se identificar a causa [12, 13];
- **Custos associados à monitorização do processo** - variáveis de qualidade que não podem ser medidas em tempo real devido à falta de sensores ou devido ao elevado valor do sensor, são muitas vezes medidas em laboratório. A utilização de sensores virtuais monitorizam essas variáveis, podendo representar uma redução de custos para a indústria.

2.2 Fases de Desenvolvimento de um Sensor Virtual

Na figura 2.1 estão ilustradas as tarefas principais no processo de desenvolvimento de um sensor virtual. A primeira tarefa, apresentada na subsecção 2.2.1, é a aquisição de dados que envolve a recolha de dados e, posteriormente, a sua análise. A etapa seguinte é discutida na subsecção 2.2.2, o pré-processamento de dados. Os métodos de seleção de variáveis são apresentados na subsecção 2.2.3. De seguida, nas subsecções 2.2.4 e 2.2.5, apresentam-se as fases de seleção, treino e validação do modelo de regressão. A última subsecção 2.2.6, é a manutenção do modelo desenvolvido.

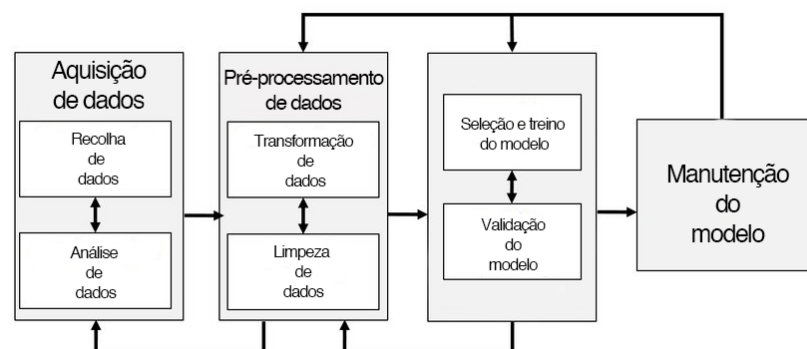


Figura 2.1: Fases de Desenvolvimento de um SS [6]

2.2.1 Recolha e Análise de Dados

Normalmente, as indústrias guardam os dados obtidos a partir dos seus processos, para depois procederem à análise e optimização dos seus componentes.

2. Visão Geral dos Sensores Virtuais

Na recolha de dados, há um fator importante a ter em consideração, a seleção da frequência de amostragem. O objetivo passa por transportar a informação dos dados do processo para um modelo preditivo, então a frequência de amostragem não pode ser demasiado baixa, de maneira a que se tenha a perceção das dinâmicas do processo. Por outro lado, este valor não pode ser muito elevado já que pode prejudicar involuntariamente o desempenho do sensor virtual.

Com a ajuda de um operador humano, faz-se uma análise aos dados recolhidos. Na existência de dados corrompidos, dados que não transportem informação útil para a análise, são removidos, evitando serem utilizados posteriormente na fase de aprendizagem do modelo.

2.2.2 Pré Processamento de Dados

O conjunto de dados recolhido em aplicações industriais reais tem, naturalmente, problemas com os quais se tem que lidar antes de serem efetivamente processados pelo modelo de aprendizagem. Para tal, são utilizadas técnicas de transformação de dados com unidades e escalas diferentes, de filtragem e deteção de *outliers* e de *missing data* [13].

A fase de pré processamento de dados normalmente divide-se em transformação de dados e limpeza de dados [6].

2.2.2.A Transformação de Dados

Muito frequentemente, os dados recolhidos envolvem variáveis que não partilham nem da mesma unidade de medida nem da mesma escala de magnitude. Por esta razão, nesta fase, os dados são frequentemente transformados em formas mais adequadas para que possam ser utilizados de forma mais eficiente pela fase da modelagem, aumentando a precisão, rapidez e a convergência de treino do modelo. Este procedimento é conhecido como normalização ou dimensionamento. As duas técnicas de normalização mais conhecidas são a normalização *min-max* e *z-score* [14].

A normalização *min-max* dimensiona e mapeia os dados originais para um intervalo específico, na qual considera que o x_{min} e x_{max} são o mínimo e máximo, respetivamente, da variável x , não dimensionada, com N amostras, onde cada amostra é representada como x_n ($n = 1, \dots, N$). Esta normalização dimensiona cada amostra de x_n no range $[x'_{min}, x'_{max}]$ pelo seguinte cálculo:

$$x'_n = \frac{x_n - x_{min}}{x_{max} - x_{min}}(x'_{max} - x'_{min}) + x'_{min}, \quad n = 1, \dots, N. \quad (2.1)$$

A normalização *z-score* ou (*zero-mean*) transforma os dados originais de acordo com a sua média e o seu desvio padrão. Dimensionar variáveis com esta técnica irá resultar em dados transformados com média nula e desvio padrão igual a um. O objetivo é transformar todos os dados para a mesma ordem de grandeza. A variável normalizada x'_n é calculada da seguinte

maneira:

$$x'_n = \frac{x_n - \bar{x}}{\sigma_x}, \quad n = 1, \dots, N \quad (2.2)$$

onde \bar{x} e σ_x são a média e o desvio padrão da variável x , respetivamente.

2.2.2.B Limpeza de Dados

A limpeza de dados é uma fase importante no desenvolvimento do sensor virtual, já que localiza e corrige problemas com algumas observações do conjunto de dados, melhorando a qualidade e consistência da informação extraída dos dados. Nesta secção, vão ser apresentados dois problemas frequentes relacionados com dados provenientes de processos industriais: dados em falta e dados inconsistentes.

Dados em Falta (*Missing Data*)

Em praticamente todos os processos industriais, a falta de dados é um problema que todos partilham. É comum não haver observações registadas num certo momento para uma ou mais variáveis. Este problema pode ser gerado por diversos fatores, por exemplo, falha no sensor físico que faz a leitura em tempo real de uma variável, momento de manutenção ou de remoção do sensor, implicando a não chegada da informação à base de dados, ou até no próprio acesso à base de dados. Existem inúmeras possibilidades para provocar que uma variável tenha falta de dados.

Se houver uma amostra em falta, numa qualquer variável, a abordagem List-Wise Deletion [15] remove a amostra em falta, para todas as variáveis, apagando toda a informação daquele instante temporal. É vantajosa ser usada se o número de valores em falta for reduzido, mas em caso contrário deve ser evitado. Outro tipo de abordagem é a imputação múltipla na qual se incluem dois métodos, a imputação por média/mediana e o *hot-deck imputation* [16]. No primeiro, o valor em falta da variável x_n é substituído pela média/mediana de x . No segundo método um valor de entrada aleatório substitui o valor em falta.

O método Maximum Likelihood (ML) modela as variáveis em falta baseando-se nos dados disponíveis. Usando relações lineares em [17] e não-lineares em [18] entre os dados disponíveis e os dados em falta, os autores verificam uma melhoria significativa em comparação com os métodos de imputação múltipla. O método Expectation-Maximization (EM) apresentado em [19] tem demonstrado ser um caminho com bons resultados e com habilidade para lidar com a questão de falta de dados.

Deteção de Dados Inconsistentes (*Outliers*)

Outliers são observações que divergem do intervalo expectável das variáveis. Afastam-se da maioria dos dados, tornando-se inconsistentes, o que conseqüentemente, se não forem detetados e eliminados, propagarão o erro e prejudicarão o desempenho dos algoritmos do mo-

2. Visão Geral dos Sensores Virtuais

delo preditivo. O mau funcionamento e a degradação do sensor físico, os erros experimentais ou mudanças não controláveis nas condições experimentais, são alguns fatores que levam à presença de *outliers*.

Existem dois tipos de *outliers*: univariados e multivariados [1]. *Outliers* univariados conseguem ser detetados pela observação de uma distribuição de valores num único espaço dimensional. *Outliers* multivariados podem ser detetados em n -espaços dimensionais e, por vezes, torna-se muito complicado para o operador humano detetá-los.

Os métodos de deteção de *outliers* univariados baseiam-se em técnicas de estatística. O algoritmo $3\sigma_x$ -rule é a abordagem mais simples de deteção de outliers univariados [1], e assume que os dados seguem uma distribuição normal. De acordo com este algoritmo, os valores x_n que satisfaçam a condição,

$$|x_n - \bar{x}| > 3\sigma_x, \quad n = 1, \dots, N, \quad (2.3)$$

são considerados *outliers*. \bar{x} e σ_x são a média e o desvio padrão da variável x (com N amostras). Em cenários em que os dados são muito afetados por *outliers* e que estes tendam a alterar muito o valor do desvio padrão, o *Hampel Identifier* (HI) é uma abordagem muito adequada [1, 6]. HI é uma versão melhorada do $3\sigma_x$ -rule na qual substitui a média pela mediana e o desvio padrão pelo desvio absoluto da mediana (*Median Absolute Deviation* - *MAD*). O desvio absoluto da mediana de uma variável x é calculada da seguinte forma:

$$MAD_x = 1.4826 \times \text{median}\{|x_n - x^*|\}, \quad n = 1, \dots, N, \quad (2.4)$$

onde o coeficiente 1.4826 é escolhido de maneira que o MAD esperado corresponda ao desvio padrão σ_x para dados com distribuição normal. x^* é a mediana da variável x . De acordo com este algoritmo, os valores x_n que satisfaçam a condição

$$|x_n - x^*| > 3 \times MAD_x, \quad n = 1, \dots, N \quad (2.5)$$

são considerados *outliers*.

Numa visão ainda mais robusta esta abordagem pode ser aplicada ao longo de uma janela deslizante. O *Hampel Identifier with Sliding Window* (HI-SW) foca-se num intervalo de dados específico e não toma o conjunto de dados como um todo. Esta abordagem tem assim 2 parâmetros: o comprimento da janela de dados e o *threshold* que define se uma determinada observação é ou não *outlier*. No caso do HI, o *outlier* é substituído pelo valor da mediana da variável x , já o HI-SW substitui o *outlier* pela mediana da janela [20].

Os métodos de deteção de *outliers* multivariados têm em consideração as interações entre variáveis. Podem ser métodos baseados em medidas de distância que mostram as amostras que estão longe do centro de distribuição de dados, como a Distância de Mahalanobis [21]. As observações com grande valor de distância Mahalanobis são consideradas *outliers* [22]. Existe

outro tipo de métodos baseados em redução da projeção ou dimensionalidade dos dados. As técnicas mais conhecidas são a *Principal Component Analysis* (PCA) e o *Partial Least Squares* (PLS). No livro de Fortuna [6] são descritas e comparadas várias técnicas usadas para detecção de *outliers* num conjunto de dados reais provenientes de uma unidade de recuperação de enxofre de uma refinaria. Os resultados obtidos mostram que a remoção dos *outliers* com as técnicas consideradas, melhoraram a capacidade de modelagem e preditiva do sensor virtual.

2.2.3 Seleção de Variáveis

Nesta secção vão ser apresentadas quatro maneiras diferentes de se fazer seleção de variáveis. A figura seguinte 2.2 refere o nome de cada um dos tipos de métodos que vão ser abordados.



Figura 2.2: Tipos de métodos de seleção de variáveis

A complexidade do modelo depende diretamente do número de variáveis de entrada que contém. Em aplicações industriais é frequente existir um elevado número de candidatos para variáveis de entrada provenientes da estrutura de supervisão do processo. A seleção de variáveis entra aqui com o papel de localizar as que verdadeiramente têm uma boa contribuição para o modelo de aprendizagem, aumentando a capacidade preditiva do mesmo, reduzindo a complexidade [23]. Nas últimas décadas, têm sido reportadas abordagens de seleção de variáveis para as várias aplicações de sensores virtuais. Estas técnicas podem ser divididas em métodos de filtro, métodos *wrapper*, métodos embebidos e métodos híbridos [24].

Nos métodos de filtro, o processo de seleção do subconjunto de variáveis acontece antes do processo de aprendizagem do modelo, portanto são métodos independentes do modelo usado. Usam critérios, medidas de estatística para avaliar o conjunto de variáveis e selecionar as mais relevantes tendo em conta um *threshold* definido inicialmente. O coeficiente de correlação (CC) e a informação mútua (IM) são as métricas de avaliação mais usadas [1]. Para regressões lineares, a correlação de Pearson é a mais conhecida para medir a correlação linear entre duas

2. Visão Geral dos Sensores Virtuais

variáveis, x e y [1]. O coeficiente de correlação de Pearson é calculado pela equação (2.6).

$$\rho(x,y) = \frac{\text{cov}(x,y)}{\sigma_x \sigma_y}, \quad (2.6)$$

onde $\text{cov}(x,y)$ é a covariância entre x e y , e σ_x e σ_y são os desvios padrões de x e y , respectivamente. O coeficiente de correlação de Pearson representado por $\rho(x,y)$ assume valores entre -1 e 1. Se $\rho(x,y) = -1$ significa que x e y têm uma correlação negativa perfeita, isto é, se uma aumenta a outra diminui. Se $\rho(x,y) = 1$, x e y têm correlação positiva perfeita. No caso de $\rho(x,y) = 0$, x e y não dependem linearmente uma da outra.

Nos métodos *wrapper*, ao contrário do métodos de filtro, o processo de seleção de variáveis está ligado ao algoritmo de aprendizagem (modelo) escolhido, ou seja, estes métodos são dependentes do modelo utilizado. São mais eficientes que os métodos de filtro, no entanto são computacionalmente mais complexos visto que todas as vezes que um subconjunto de variáveis tem que ser avaliado tem que se treinar e testar o modelo de regressão. Alguns exemplos destes métodos são os algoritmos sequenciais: *Forward Selection*, *Backward Selection* e *Stepwise Selection* [25].

Os métodos *embedded* incluem na sua metodologia a capacidade de selecionar o melhor subconjunto de variáveis enquanto treinam e avaliam o modelo. Alguns métodos de regularização, como o *Least Absolute Shrinkage and Selection Operator* (LASSO) e o *Elastic Net* (EN), fazem parte deste tipo de métodos, já que possuem a capacidade interna de remover variáveis de entrada durante o processo de treinamento do modelo. Os métodos referidos introduzem um termo de regularização nos coeficientes de regressão, sendo alguns reduzidos ao valor nulo. Nesta situação, essas variáveis deixam de ser consideradas como variáveis de entrada para o modelo. Apesar dos métodos *embedded* partilharem características similares aos métodos *wrapper*, os *embedded* não podem ser usados em combinação com outro modelo [26].

Os métodos *hybrid* combinam diferentes algoritmos com o objetivo de promover a seleção de variáveis de entrada. As abordagens *hybrid* combinam métodos de filtro com métodos *wrapper*. Os métodos de filtro fazem uma seleção preliminar ao conjunto de variáveis de entrada disponíveis, ficando as que apresentarem maior *score* tendo em conta a sua relevância para a variável de saída. Após esta primeira etapa, os métodos *wrapper* investigam o subconjunto que permita ao modelo de regressão obter o melhor desempenho possível. A função dos métodos de filtro é muito importante, já que diminuem a complexidade computacional imposta aos *wrapper*, uma vez que reduzem o número de avaliações dos subconjuntos. A combinação de métodos resulta numa elevada eficiência e precisão. Algumas metodologias interessantes foram propostas no artigo [24], tais como a *fuzzy random forest* baseado em seleção de variáveis e *hybrid genetic algorithms*.

2.2.4 Seleção e Treino do Modelo

O sensor virtual pode ser desenvolvido através de 3 categorias diferentes de modelos [1]: *white-box* (*model-driven*), *gray-box* e *black-box* (*data-driven*), como mostra a figura 2.3. A seleção do tipo de modelo está frequentemente sujeita à preferência pessoal, à experiência passada dos desenvolvedores e à informação disponível do ambiente em que está inserido.

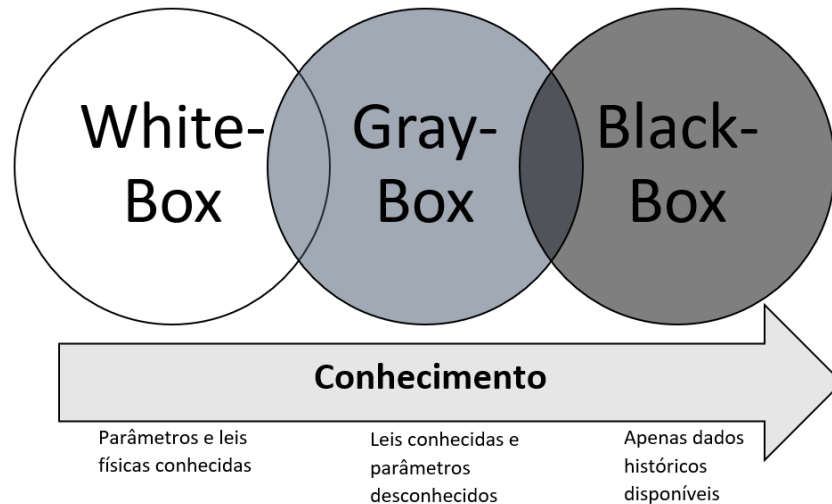


Figura 2.3: Abordagens de Modelagem do SS

O desenvolvimento do sensor virtual baseado em modelos *white-box*, requer forte conhecimento físico e químico dos sistemas e processos. São descritivos e de fácil interpretação por parte do operador industrial no entanto, em sistemas mais complexos esta abordagem pode não ser adequada, já que por vezes não se conseguem entender as dinâmicas atuais do processo, aumentando a dificuldade de modelação, provocando consequentemente uma redução do desempenho predictivo.

Com o avançar da indústria, a maioria dos processos industriais são cada vez mais complexos e descrevê-los em termos de equações físicas e químicas tem-se tornado num grande desafio, necessitando-se de um grande conhecimento do processo. Por esta razão, os modelos *black box*, ou *data-driven*, têm sido uma solução confiável, mostrando resultados muito interessantes em aplicações de sensores virtuais. Não necessita de conhecimento *a-priori* do processo, mas através de informação extraída a partir de dados históricos do processo desenvolve os seus modelos predictivos. A necessidade de dados de qualidade e a baixa intuição para os operadores humanos são limitações deste tipo de modelos [14]. Dentro deste modelo existem duas opções que se podem usar: um modelo linear ou um modelo não-linear. Segundo o artigo [7], os modelos lineares mais usados são os *Partial Least Squares* (PLS), *Principal Component Analysis* (PCA) em combinação com um modelo de regressão, *Least Absolute Shrinkage and Selection Operator* (LASSO) e *Ridge Regression* (RR). Para modelos não-lineares é muito comum serem

2. Visão Geral dos Sensores Virtuais

usadas as *Artificial Neural Networks* (ANNs), *Support Vector Machine* (SVM) e *Neuro-Fuzzy System* (NFS) [27,28].

Os PLS focam-se na relação entre as variáveis de entrada e saída a partir da matriz de covariância, mantendo um bom desempenho para dados colineares, característica que é bastante recorrente em dados industriais, enquanto cumpre as restrições de ortogonalidade. É um método interessante quando o número de variáveis de entrada é elevado. Em 1998, surgiu o método *Recursive Partial Least Squares* (RPLS) [29]. O modelo é atualizado para refletir o estado atual do processo enquanto novas amostras vão surgindo. Em 2019, [30] aplicou um sensor virtual, usando o método PLS, para prever a quantidade de gás numa secção de numa mina de carvão, e obtiveram-se-se bons resultados de previsão com um rápido tempo de treino do modelo. Recentemente, desenvolveu-se um modelo inferencial baseado em PLS capaz de prever com alta precisão a concentração de amónio [31].

Em casos em que a dimensionalidade do espaço de entrada é elevada e há presença de multicolinearidade, a RR [32] e LASSO [33] são métodos interessantes. No artigo [34], o RR e LASSO demonstraram-se ser muito eficientes na previsão de variáveis de qualidade numa refinaria de petróleo e na manutenção preditiva de microturbinas a gás.

Quanto aos modelos *data-driven* de sistemas não-lineares, as redes neuronais artificiais (ANNs) são dos modelos mais conhecidos. O autor do artigo [35] dá conta que a sua história começou quando Warren McCulloch e Walter Pitts inventaram a ANN [36], e que durante o século 20, as ANNs foram alvo de uma grande investigação e aplicadas em diferentes áreas. Em 1986, o algoritmo de aprendizagem *backpropagation* da ANN ficou popularizado [37]. Define-se uma rede neuronal artificial composta por uma camada de entrada, uma ou mais camadas ocultas e uma camada de saída [38]. Com o intuito de funcionar tal como as redes neuronais biológicas, as ANNs possuem nós ou unidades de processamento que extraem informação das variáveis de entrada (*input layer*). Estas unidades de processamento que representam os neurónios biológicos processam essa informação e propagam-na para outras unidades (*hidden layers*). A camada final (*output layer*) conclui e apresenta uma resposta de saída. A topologia da rede neuronal também é uma escolha difícil de ser feita, já que se tem que testar inúmeros casos para que se obtenha o melhor desempenho possível [38]. A *MultiLayer Perceptron* (MLP) com o algoritmo de aprendizagem *backpropagation* é uma das ANNs mais utilizadas [39]. Este algoritmo é simples de se implementar e computacionalmente eficiente: no entanto uma grande limitação do algoritmo é que ele pode ser extremamente lento, principalmente quando temos que lidar com uma tarefa de aprendizagem difícil que requer o uso de uma grande rede [40]. Associado às ANNs, temos técnicas de otimizar os seus hiper-parâmetros, como por exemplo o *Genetic Algorithm* (GA), e *Simulated Annealing* (SA). No ano 2000, surgiu uma ANN com o algoritmo de aprendizagem *Extreme Learning Machine* (ELM) [41], com um tempo de treino mais rápido e um melhor desempenho do que a tradicional ANN [42].

O método de aprendizagem supervisionada *Support Vector Machine* (SVM) tem demonstrado ser também uma das técnicas com mais sucesso em ML nos últimos anos. O artigo [43] reporta exemplos de aplicações de SVM em projetos de previsão de variáveis na indústria do petróleo e de mineração. A implementação do método passa pela seleção da função *kernel* e de outros parâmetros dos SVMs, no entanto, pela revisão da literatura conclui-se que a função *kernel-radial basis* (RBF) é a mais eficiente de todas as disponíveis. É preciso referir que, apesar das boas abordagens do SVM, este método pode pesar computacionalmente se a dimensão do conjunto de dados for muito grande.

Os modelos *gray-box* também conhecidos como modelos híbridos, integram modelos *white-box* e *black-box*. O desenvolvimento do modelo incorpora tanto o conhecimento *a-priori* do processo como também os dados experimentais [44]. Em [45], o autor faz uma revisão a temas como o desenvolvimento do modelo *gray-box* e apresenta alguns exemplos de aplicações em processos industriais.

2.2.5 Validação do Modelo

Esta etapa tem como objetivo avaliar o desempenho do modelo treinado, isto é, testa a capacidade preditiva do modelo em amostras novas, que não foram usadas durante a fase de treino. Nesta secção apresentam-se metodologias que são usadas para comparar os resultados dos modelos de regressão.

Dependendo da quantidade de dados disponíveis existem diferentes abordagens que podem ser usadas. Se o conjunto de dados for elevado, o que é feito tipicamente é a divisão aleatória ou manual dos dados em dois conjuntos, um para treino e outro para teste [1]. Treina-se o modelo usando os dados de treino e no fim, usando os dados de teste, avalia-se o desempenho do modelo usando algumas métricas, entre as quais se destaca, o *mean squared error* (MSE), o *root mean squared error* (RMSE), o *mean absolute error* (MAE) e o coeficiente de determinação (R^2). Se a quantidade de dados disponível for baixa, normalmente usam-se técnicas de validação cruzada [46], como por exemplo a *K-fold Cross-Validation* (K-F CV) e a *Leave-One-Out Cross Validation* (LOOCV).

O método *K-Fold cross-validation* divide aleatoriamente o conjunto de dados em subconjuntos de dados (*k*-folds). Da divisão, $k - 1$ -folds são usados para treinar o modelo e o *k*-fold, que não se usou no processo de treino, avalia o desempenho do modelo. Este processo repete-se para todos os *k*-folds e no fim calcula-se a média da métrica de desempenho. Na Figura 2.4 apresenta-se o exemplo de uma validação cruzada de 5-folds.

2. Visão Geral dos Sensores Virtuais

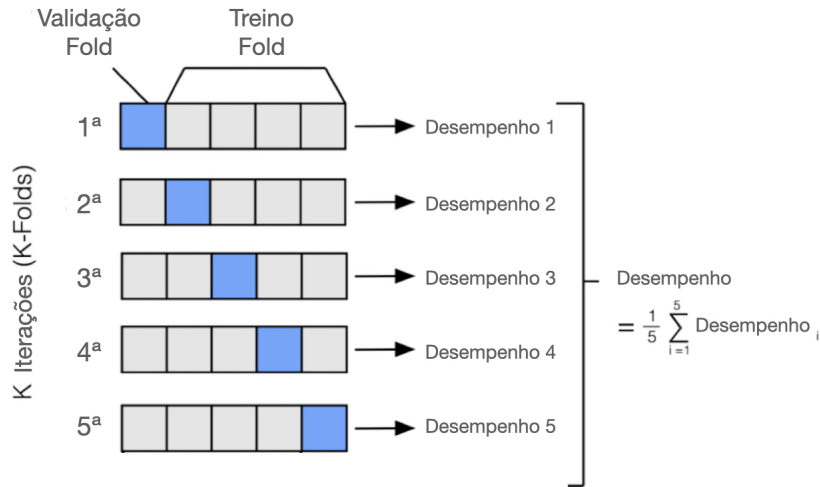


Figura 2.4: *5-fold cross-validation* [47]

Quando k é igual ao número de observações disponíveis estamos perante um caso especial de validação cruzada k -fold, a *Leave-one-out Cross Validation* (LOOCV). O modelo é treinado usando $N - 1$ observações, enquanto uma única observação é usada para prever a resposta. Este processo repete-se N vezes, obtendo-se N valores da métrica de desempenho (RMSE, MSE, MAE, entre outras). No final calcula-se a média das N métricas obtidas ao longo do processo. Na Figura 2.5 é apresentado o procedimento desta técnica, onde se pode verificar que um conjunto de dados constituído por N observações é repetidamente dividido num conjunto de treino com $N - 1$ amostras (assinaladas a azul) e num conjunto de teste contendo uma única observação (assinalado a bege) [48].

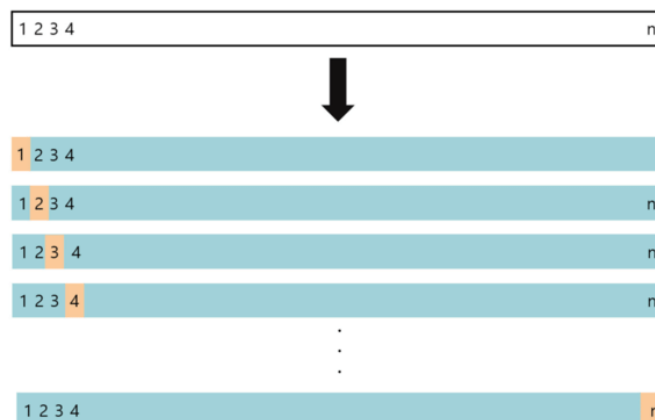


Figura 2.5: *Leave-one-out Cross Validation*

2.2.6 Manutenção do Sensor Virtual

Após o desenvolvimento e a implementação do sensor virtual, é necessária uma manutenção regular visto que existem vários fatores capazes de alterar o comportamento normal do sensor. Os processos industriais variam ao longo do tempo e os dados históricos podem deixar de conseguir representar o processo e, por estas razões, há a probabilidade de o sensor virtual vir-se a degradar e o seu desempenho piorar [49]. Desta forma, pode-se atualizar o modelo de forma online ou recursiva com novas amostras do processo. Na revisão feita em [19] são apresentadas metodologias para o desenvolvimento de sensores virtuais adaptativos. Algumas metodologias adaptativas, como janela deslizante (*sliding window* - SW), diferença temporal e regressão localmente ponderada (*locally weighted regression* - LWR) foram propostas e aplicadas em contexto industrial.

3

Modelos de Regressão Preditivos Aplicados aos Sensores Virtuais

Conteúdo

3.1	Introdução	22
3.2	Multiple Linear Regression	23
3.3	Métodos de Seleção de Variáveis	24
3.4	Métodos de Regressão de Penalização	26
3.5	Métricas de Desempenho Adotadas na Tese	31
3.6	Comparação dos Modelos	32

3.1 Introdução

Como a literatura recente está composta por inúmeros modelos preditivos que podem ser considerados no desenvolvimento dos sensores virtuais, esta dissertação vai-se focar em 7 modelos de regressão. Na Figura 3.1 estão apresentados os modelos de regressão usados.

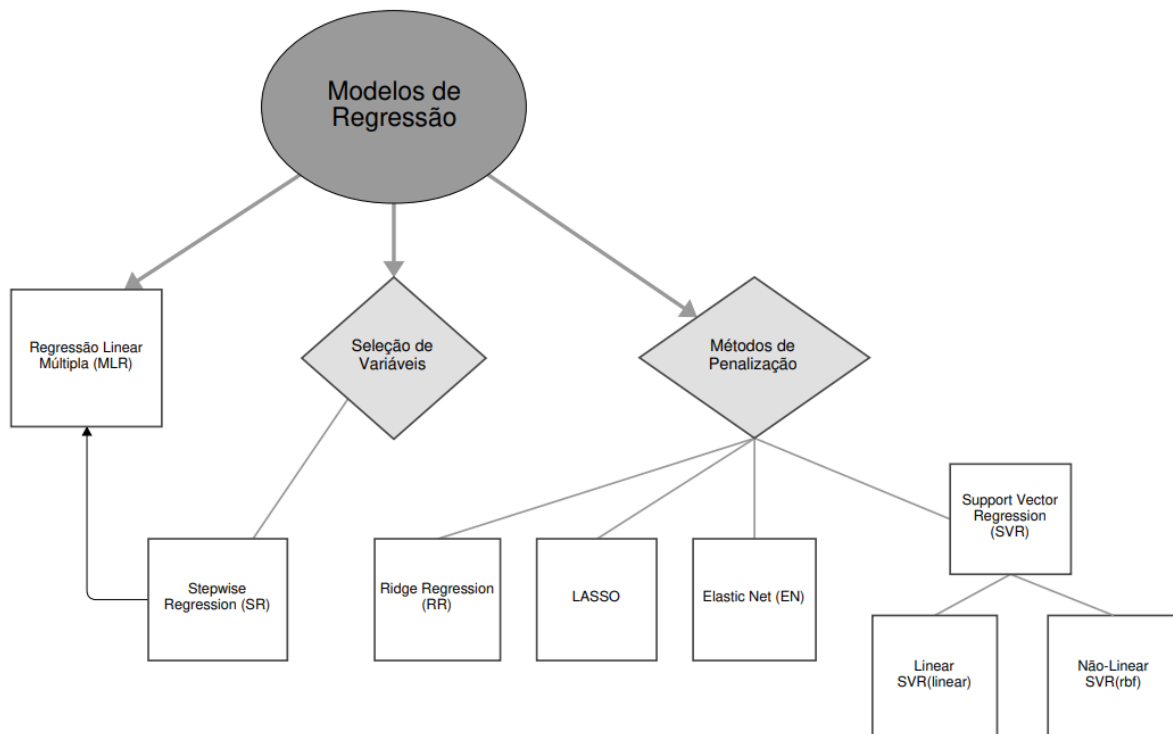


Figura 3.1: Modelos de Regressão aplicados aos SSs

O capítulo está organizado da seguinte forma. Inicialmente, na Secção 3.1.1 é descrito o problema de regressão. Na Secção 3.2 é apresentada a regressão linear múltipla (MLR). A Secção 3.3 apresenta a *Stepwise Regression* (SR), que está incluída na categoria de modelos de seleção de variáveis. Na Secção 3.4 são descritos modelos de regressão de penalização, na qual incluímos o *Ridge Regression* (RR), o *Least Absolute Shrinkage and Selection Operator* (LASSO), o *Elastic Net* (EN) e o *Support Vector Regression* (SVR). As métricas usadas para avaliar o desempenho preditivo de cada um dos modelos de regressão são apresentadas na Secção 3.5. Na Secção 3.6 é explicada a metodologia usada para comparar os modelos preditivos usados.

3.1.1 Problema de Regressão

O processo de treino é feito através do uso de pares de entradas-saídas conhecidos, sendo classificada como aprendizagem supervisionada. Para N amostras da forma $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_N, y_N)\}$, cada entrada x_n , da amostra n , é representada na forma de um

vetor $\mathbf{x}_n = [x_n^1, x_n^2, \dots, x_n^m]^N$, onde x_n^i é o valor da amostra n da i -ésima variável de entrada, e y_n o valor de saída da amostra n .

Sendo n o número de amostras e m o número de variáveis de entrada, o principal objetivo da regressão é obter uma função $f : \mathbb{R} \rightarrow \mathbb{R}^m$, baseada num conjunto de dados de treino $D = \{(\mathbf{x}_n, y_n)\}_{n=1}^N$, e recorrendo a N amostras de treino, capaz de prever o valor real do vetor de saída $\mathbf{y} \in \mathbb{R}^n$ com base num conjunto de dados de entrada $\mathbf{x} \in \mathbb{R}^{n \times m}$. Na Equação 3.1 está representada a matriz \mathbf{x} , correspondente às amostras das variáveis de entrada, e \mathbf{y} a matriz das amostras da variável de saída.

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} x_1^1 & x_1^2 & \dots & x_1^m \\ x_2^1 & x_2^2 & \dots & x_2^m \\ \vdots & \vdots & \ddots & \vdots \\ x_n^1 & x_n^2 & \dots & x_n^m \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \quad (3.1)$$

Considerando que a saída y_n pode ser aproximada por $f(x_n, \beta)$, sendo β o vetor de parâmetros do modelo, com $\beta \in \mathbb{R}^{m+1}$, em função das variáveis de entrada x_n , da seguinte forma:

$$y_n = f(x_n, \beta) + \varepsilon_n, \quad (3.2)$$

onde ε_n é o erro residual da amostra n , $\varepsilon_n \in \mathbb{R}^n$ que assume-se ter uma distribuição normal de média nula e variância σ^2 .

O objetivo da regressão é obter um modelo preditivo com bom desempenho: contudo, quando existe a quantidade de dados disponível é baixa, ocorre o problema de *overfitting*, isto é, o modelo de regressão tem bons resultados no processo de treino, mas mostra-se ineficaz em prever estimativas para novas amostras, costumam-se utilizar técnicas de validação cruzada para limitar a ocorrência deste problema.

3.2 Multiple Linear Regression

O Multiple Linear Regression (MLR) é muito conhecido e usado nos problemas de regressão. É frequentemente utilizado conjuntamente com métodos de seleção de variáveis, graças à sua simplicidade de implementação. O modelo assume que a relação entre as variáveis é linear e que cada uma segue uma distribuição gaussiana. Assume também que os termos de erro, ε_n , têm variância constante e são independentes dos valores das variáveis independentes (também conhecidas como preditores). Na Equação (3.3) apresenta-se o modelo:

$$\hat{y}_n = \beta_0 + \sum_{i=1}^N \beta_i x_i, \quad (3.3)$$

onde \hat{y}_n corresponde à variável de saída estimada da amostra n , $\mathbf{x}_i \in \mathbb{R}^n$ representa o vetor de amostras da variável i , β_i é o coeficiente de regressão da variável i e β_0 é a interseção de y (valor de y quando todos os outros parâmetros são iguais a 0).

3. Modelos de Regressão Preditivos Aplicados aos Sensores Virtuais

Na Equação (3.4), apresentam-se os coeficientes de regressão, $\hat{\beta}_{MLR}$, que são calculados através do Método dos Mínimos Quadrados (do inglês *Least Squares* []). É uma técnica de otimização matemática que procura minimizar a soma dos quadrados das diferenças entre o valor estimado e o valor real.

$$\hat{\beta}_{MLR} = \underset{\beta=[\beta_0, \beta_1, \dots, \beta_m]}{\operatorname{arg\,min}} \left\{ \sum_{n=1}^N (y_n - \hat{y}_n)^2 \right\} \quad (3.4)$$

onde N é o número de amostras, y_n é a n -ésima resposta observada e \hat{y}_n a respetiva resposta estimada.

Num conjunto de dados com elevado nível de colinearidade entre as variáveis independentes, a variância dos coeficientes de regressão aumenta, tornando-os instáveis, acabando por dificultar a sua interpretação.

3.3 Métodos de Seleção de Variáveis

Métodos de seleção de variáveis são desenvolvidos com o intuito de identificarem o subconjunto de variáveis de entrada que melhor descreve a variável resposta. Estes métodos removem preditores redundantes (que partilhem informação com outros preditores), podem evitar *overfitting*, reduzem o tempo de computação e melhoram o desempenho preditivo do modelo de regressão [50].

Esta secção descreve o algoritmo utilizado para selecionar as variáveis de entrada, o *Stepwise Regression* método que trabalha em conjunto com o MLR.

3.3.1 Stepwise Regression

O objetivo geral do desenvolvimento de um modelo de regressão é reduzir a soma do erro quadrático (SSE) ao menor valor possível. A *Stepwise Regression* (SR) é uma abordagem iterativa, incluída nos métodos de seleção *wrapper*, já que seleciona o subconjunto de variáveis de entrada que vai ser usado no final do modelo para prever a variável de resposta, dependendo da eficiência do algoritmo de aprendizagem que se use.

Este algoritmo é a junção de dois métodos de seleção de variáveis, o *forward selection* e o *backward elimination*. No *forward selection* o modelo começa sem qualquer variável, e em cada iteração é-lhe adicionada a variável mais significativa, tendo em conta um critério pré-definido, como testes estatísticos (por exemplo, *p-value*, que foi o critério utilizado) ou outras técnicas de avaliação como o R^2 ajustado, critério de informação Aikake, critério de informação Bayesiano e o Mallows' Cp. Quando adicionada ao modelo, a variável que apresentar o menor *p-value* e que esteja abaixo do *threshold* pré-definido, ou em casos que variáveis tenham o mesmo valor de *p-value*, a que fornecer um maior aumento no valor do coeficiente de determinação ajustado, R_{adj}^2 , é considerada a variável mais significativa e é adicionada ao subconjunto final.

Em *backward elimination* o modelo começa com todas as variáveis disponíveis e faz o processo inverso da *forward selection*, vai eliminando a variável menos significativa em cada iteração. O processo de seleção está terminado quando não houver variáveis com *p-value* abaixo do valor de *threshold*, ou seja, que não melhorem o ajuste do modelo.

Na Figura 3.2 estão descritos, na forma de um diagrama, os passos que o SR executa ao longo do seu procedimento de seleção das variáveis de entrada.

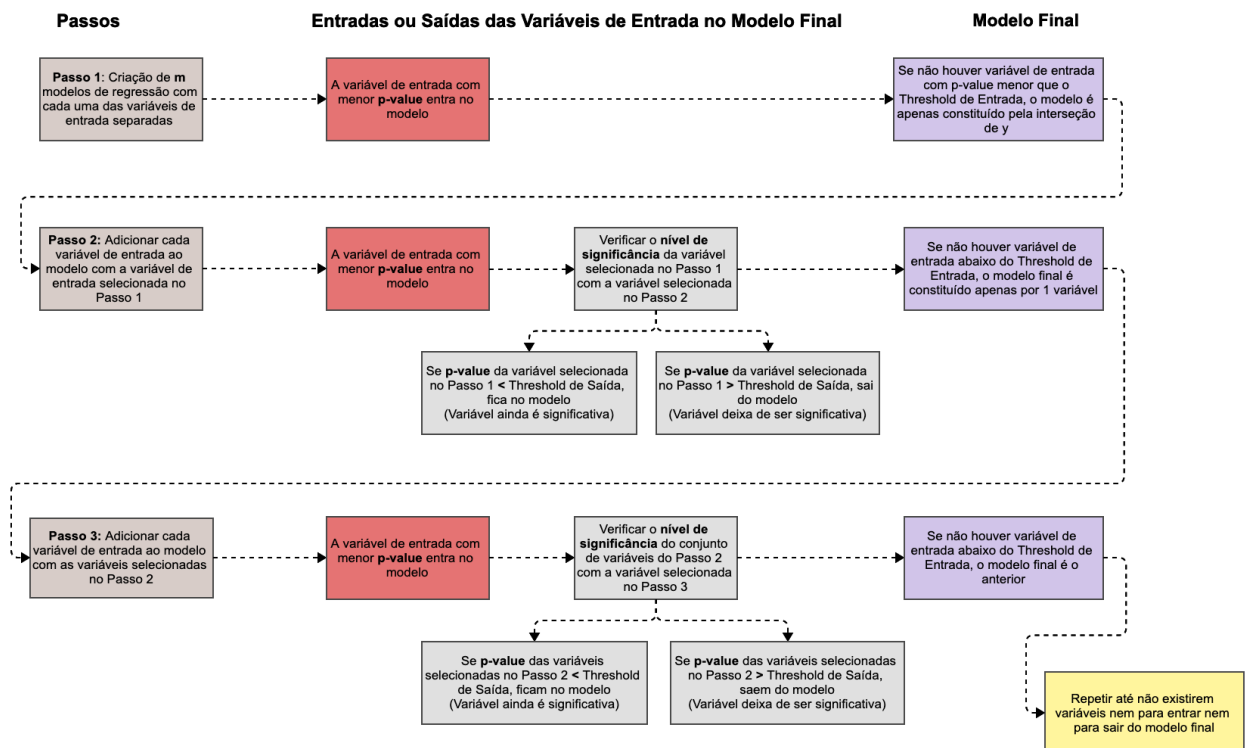


Figura 3.2: Procedimento do Stepwise Regression

No método SR, existem dois valores de *threshold*, um para a entrada (*threshold-in*) e outro para a saída (*threshold-out*) da variável de entrada. O algoritmo começa por seleccionar a variável mais significativa dentro do critério explicado anteriormente, ou seja, criam-se m modelos de regressão linear, cada um correspondendo a uma variável de entrada e comparam-se os p -value obtidos por cada uma, a que tiver menor valor e estiver abaixo do *threshold-in* entra no subconjunto de variáveis seleccionadas (Passo 1). Repete-se o processo agora construindo $m - 1$ modelos de regressão linear, compostos pela variável inicialmente seleccionada e com cada uma das restantes variáveis. As variáveis candidatas passam pelo critério de avaliação relacionado com o seu p -value, e a melhor variável dentro desse critério entra no subconjunto. Há medida que as variáveis vão entrando no modelo, a capacidade individual de explicar a variável de resposta vai sofrendo alterações e o seu p -value inicial pode tomar outro valor. Se esse novo p -value ultrapassar o *threshold-out*, a variável deixa de ser significativa para o modelo e

3. Modelos de Regressão Preditivos Aplicados aos Sensores Virtuais

é removida do mesmo (Passo 2). De notar também que, ao contrário das duas abordagens de seleção explicadas no início, na *Stepwise Regression*, tanto as variáveis que entram como as que saem podem voltar a ser integradas no modelo, numa das iterações seguintes, claro, se o seu *p-value* cumprir com o critério pré-definido. Este processo de seleção repete-se até que o critério convirja e não sofra mais alterações (Passo 3).

3.4 Métodos de Regressão de Penalização

Em casos em que há presença de *overfitting* no modelo de regressão, os métodos de regularização podem servir de solução para resolver este problema, já que impõem penalidades aos coeficientes de regressão, forçando que, variáveis menos contributivas, tenham o seu coeficiente próximo ou igual a zero, diminuindo a variância e a sua complexidade e, conseqüentemente, que se atinjam bons resultados na estimação da variável de saída.

Existe uma relação *trade-off* entre a variância e o *bias*. De notar que, a variância mede a sensibilidade de estimativa em relação ao conjunto de dados, e o *bias* mede a diferença entre a previsão esperada e o valor real da saída. Esta relação não é simples, já que reduzindo um termo, aumenta-se o valor do outro, portanto o objetivo é arranjar o melhor equilíbrio entre a variância e o *bias* [51]. Um modelo com elevada variância terá bons resultados nos dados de treino, mas fracos com novas amostras. Um modelo com *bias* elevado é um modelo muito simples, em termos de complexidade, e com pouca precisão. Na Figura 3.3 apresenta-se a influência que a variância e o *bias* têm na complexidade do modelo e no erro total de previsão.

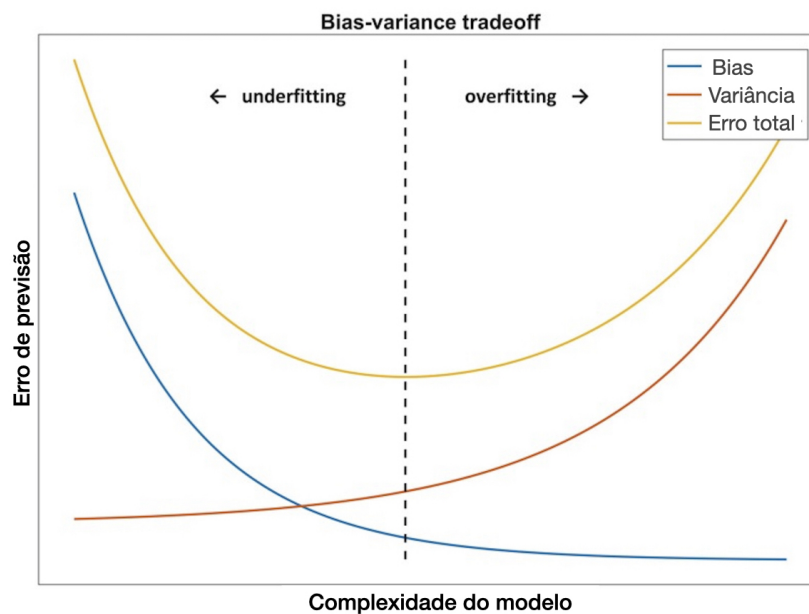


Figura 3.3: Complexidade do modelo vs Erro de previsão [52]

Pela Figura 3.3 é observado que modelos com complexidade reduzida (modelos simples,

caracterizados com poucos parâmetros, com poucas variáveis no espaço de entrada) têm um valor de variância baixo mas um elevado valor em termos de bias. Estes modelos sofrem de *underfitting*, não conseguem extrair padrões a partir dos dados, obtendo um fraco desempenho preditivo e, conseqüentemente, um elevado valor de erro de previsão. Por outro lado, modelos com uma maior complexidade, sofrem do fenômeno de *overfitting*, na qual o seu bias apresenta valores mínimos, mas o valor de variância apresenta níveis elevados. Tendo em conta estes dois casos extremos, o objetivo passa por encontrar a complexidade ideal para que se obtenha o erro total mínimo, não entrando nem no campo de *underfitting*, nem no de *overfitting*, por isso está representado a linha tracejada vertical o nível de complexidade do modelo que cumpre essas condições.

Nesta categoria de métodos apresenta-se o *Ridge Regression* (RR), o *Least Absolute Shrinkage and Selection Operator* (LASSO), o *Elastic Net* (EN) e o *Support Vector Regression* (SVR), que implementam quatro maneiras diferentes de restringir os pesos dos coeficientes de forma a que se reduza a variância e aumente o *bias* do modelo.

3.4.1 Ridge Regression

O *Ridge Regression* (RR) é uma versão regularizada da regressão linear, visto que é adicionada à função de custo um termo de regularização (l_2) que força o algoritmo de aprendizagem a ajustar os dados e, ao mesmo tempo, torna os pesos do modelo o mais pequeno possíveis.

A quantidade de regularização aplicada ao modelo é controlada pelo hiper-parâmetro λ . Se λ tem valor 0, a equação é a mesma que o método dos mínimos quadrados e então, o RR é uma regressão linear comum. Se aumentar o valor de λ , o impacto da penalização aumenta e os coeficientes do RR tendem a aproximar-se de zero. A *Ridge Regression* “encolhe” (*shrinks*) os coeficientes de regressão impondo uma restrição na norma l_2 dos mesmos. O termo de regularização l_2 e a função de custo da RR são apresentados nas Equações 3.5–3.6, respetivamente [53].

$$l_2 = \lambda \sum_{i=1}^m \beta_i^2 \quad (3.5)$$

$$\hat{\beta}_{ridge} = \min_{\beta} \sum_{i=1}^n \left(y_i - \beta_0 - \sum_{j=1}^m \beta_j x_{ij} \right)^2 + \lambda \sum_{j=1}^m \beta_j^2 \quad (3.6)$$

3.4.2 Least Absolute Shrinkage and Selection Operator

O método de penalização LASSO é uma versão regularizada da regressão linear e pode ser descrita como uma minimização de uma função graças a uma restrição. O LASSO é uma versão do método dos mínimos quadrados, descrito na Equação 3.4, no entanto é adicionada à função de custo um termo de regularização (l_1), que força o algoritmo de aprendizagem a ajustar os

3. Modelos de Regressão Preditivos Aplicados aos Sensores Virtuais

dados e, ao mesmo tempo, torna os pesos do modelo o mais pequeno possíveis. Em casos de elevada correlação entre variáveis de entrada, esta penalização tem um papel fundamental, já que mantém a estabilidade da minimização do erro entre a variável observada e a variável estimada, ao mesmo tempo que mantém a convexidade da função de custo [54].

Na Equações 3.7–3.8 são apresentados o termo de regularização, l_1 , e a função de custo do LASSO, respetivamente.

$$l_1 = \lambda \sum_{i=1}^m \beta_i \quad (3.7)$$

$$\hat{\beta}_{lasso} = \min_{\beta} \sum_{n=1}^N \left(y_n - \beta_0 - \sum_{i=1}^m \beta_i x_i \right)^2 + \lambda \sum_{i=1}^m \beta_i \quad (3.8)$$

Como resultado desta penalização, ao contrário do RR, o LASSO penaliza os coeficientes de regressão, e dependendo do valor λ , variáveis que não sejam significativas para o modelo acabam por ter o seu valor nulo. Por esta razão, o LASSO possui a vantagem de reduzir coeficientes a zero, podendo ser uma alternativa para redução de dimensionalidade.

3.4.3 Elastic Net

O método de regressão *Elastic Net* (EN) proposto em [55], combina os termos de regularização das técnicas LASSO e RR. A vantagem deste modelo é permitir uma efetiva regularização com a penalização l_2 do método RR, especialmente em grupos de preditores correlacionados, juntamente com a qualidade da seleção de variáveis através do LASSO.

Os coeficientes de regressão do EN são estimados através da Equação 3.9.

$$\hat{\beta}_{EN} = \underset{\beta}{\operatorname{argmin}} \left\{ \sum_{n=1}^N (y_n - \hat{y}_n) + \lambda \left(\alpha \sum_{i=1}^m |\beta_i| + \frac{1-\alpha}{2} \sum_{i=1}^m \beta_i^2 \right) \right\} \quad (3.9)$$

O hiper-parâmetro α ($\alpha \in [0, 1]$) controla a contribuição relativa de cada tipo de penalização, l_1 e l_2 , aplicados à magnitude dos coeficientes de regressão. O hiper-parâmetro λ controla o *trade-off* entre o *bias* e a variância. Se λ for igual a zero, a solução passa a ser igual ao método mínimos quadrados, apresentado na Equação 3.4, para o método MLR. Valores baixos de λ fazem com que o modelo tenha elevada variância e reduzido *bias*. Por outro lado, um valor elevado de λ , força os coeficientes de regressão a um valor mais pequeno, resultando num modelo mais *biased*.

3.4.4 Support Vector Regression

Support Vector Regression (SVR) é uma versão da *Support Vector Machine*, relativamente a problemas de regressão [56]. A SVR pode ser linear ou não-linear dependendo da função

kernel que se use. Esta técnica é, normalmente, muito robusta em problemas não-lineares, quando o número de amostras disponíveis é reduzido. O modelo SVR estima uma função com uma restrição na qual cada valor estimado da variável resposta tem um desvio máximo de ϵ relativamente ao seu valor real. Na Figura 3.4 está representado graficamente o modelo SVR. O hiperplano e as linhas de vetor de suporte (linhas de fronteira), que contém os pontos mais próximos do hiperplano, formam o tubo (ϵ -insensitive tube) que irá conter as amostras de treino para a regressão. O objetivo do SVR é minimizar o tubo fazendo com que seja o mais estreito possível, mas que ao mesmo tempo inclua o máximo de amostras de treino. As amostras que estejam dentro do tubo e que se aproximem o mais possível da linha central do hiperplano, serão vistas como fazendo parte do cenário ideal, correspondendo ao melhor ajuste possível, ou seja, ao menor erro de previsão.

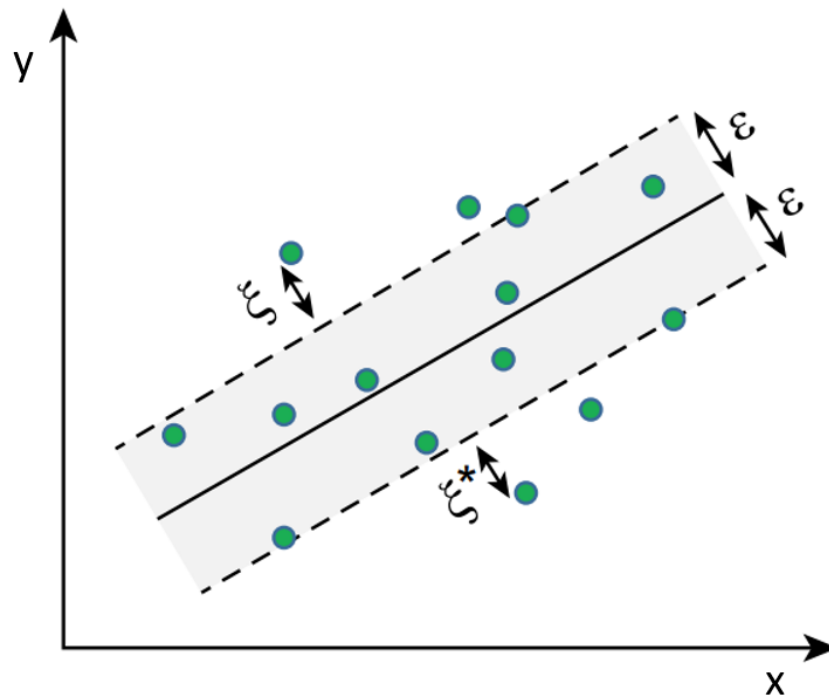


Figura 3.4: Representação do modelo SVR

Para o caso linear do SVR, a previsão de y_n é dada pela Equação 3.10.

$$\hat{y}_n = f(x_n) = \beta^T x_n + b \quad (3.10)$$

onde β é o vetor de pesos, \mathbf{x}_n é o vetor de entradas e b o *bias*.

A função de erro do SVR é dada pela Equação 3.11.

$$L = \min \frac{1}{2} \|\beta\|_2^2 + C \sum_{n=1}^N (\xi_n^v + \xi_n^a), \quad (3.11)$$

3. Modelos de Regressão Preditivos Aplicados aos Sensores Virtuais

onde o intervalo de operação é descrito de seguida:

$$-\varepsilon - \xi_n^\vee \leq y_n - \beta^T x_n - b \leq \varepsilon + \xi_n^\wedge, \text{ para } \xi_n^\vee \geq 0, \xi_n^\wedge \geq 0 \text{ (n = 1, 2, \dots, N)}.$$

O primeiro termo de L é um termo de penalização que controla a complexidade do modelo. O segundo termo corresponde à *insensitive-function* ε . Esta função ignora valores absolutos do erro, se a distância entre o valor observado, y , e a sua previsão for menor que o limite ε . Inclui um termo C , usado para penalizar os erros das amostras, e duas *slack variables*, ξ , que permitem que haja um maior intervalo de erro da variável x_i .

A função de erro ε dada pela Equação 3.12, destaca o facto de não haver perda para desvios menores que ε , e que para grandes desvios, as amostras são linearmente penalizadas.

$$L_\varepsilon = \begin{cases} 0, & \text{se } |f(x_n) - y_n| < \varepsilon \\ |f(x_n) - y_n| - \varepsilon, & \text{outro caso} \end{cases} \quad (3.12)$$

De notar que x_n é o vetor de entrada da amostra n e y_n o vetor de saída, relativamente aos dados de treino.

Como forma de minimizar a função de erro do SVR, são feitas transformações com o uso de multiplicadores de *Lagrange*, α_n^* e α_n . Na Equação 3.13 é descrita como é realizada essa otimização.

$$f(x_n) = \sum_{n=1}^N (\alpha_n^* - \alpha_n) x_n^T x + b \quad (3.13)$$

Alguns problemas de regressão podem não ser descritos adequadamente utilizando um modelo linear. A Equação de otimização de Lagrange 3.13, permite que a técnica possa ser estendida a outras funções *kernel*, transformando o espaço original de entrada, onde não se consegue fazer uma separação linear entre as amostras de treino, num espaço de alta dimensão (espaço *kernel*). A fórmula geral está apresentada na Equação 3.14.

$$f(x_n) = \sum_{n=1}^N (\alpha_n^* - \alpha_n) K(x_n^T, x) + b \quad (3.14)$$

Na Tabela 3.1 são apresentadas algumas opções de funções *kernel* que se podem usar na implementação do SVR. Na tese foram usados o *kernel* Linear e o *kernel* da Função de Base Radial (RBF). O *kernel* Polinomial não foi usado devido ao seu elevado tempo de computação.

Tabela 3.1 Funções *Kernel* e o seu campo de aplicação

Nome	Equação	Campo
Kernel Linear	$K(x_i, x_j) = x_i x_j$	SVR Linear
Kernel Polinomial	$K(x_i, x_j) = (1 + x_i x_j)^p, p = 2, 3, \dots$	SVR Não-Linear
Kernel RBF	$K(x_i, x_j) = \exp(- x_i - x_j ^2)$	SVR Não-Linear

3.5 Métricas de Desempenho Adotadas na Tese

Para avaliar o desempenho preditivo de um modelo é necessário calcular e analisar diferentes parâmetros estatísticos. O foco principal é o cálculo do erro de previsão, que corresponde à diferença entre o valor observado (y_n) e o valor estimado para a variável de saída (\hat{y}_n). As métricas usadas para medir o desempenho preditivo de cada um dos modelos, tanto no processo de treino como no processo de teste, foram: o *Root Mean Square Error* (RMSE) e o *R-squared* (R^2), também conhecido como coeficiente de determinação. Cada uma das métricas vai ser apresentada de seguida.

3.5.1 Root Mean Square Error

O Root Mean Square Error é uma métrica normalmente usada para calcular o erro de previsão obtido pelo modelo preditivo. Quanto maior for a diferença entre o valor estimado e o valor real, maior será o valor de RMSE. O objetivo é obter um valor baixo RMSE, visto que corresponde a uma boa previsão por parte do modelo. RMSE é determinado usando a expressão descrita na Equação 3.15.

$$RMSE = \sqrt{MSE} = \sqrt{\frac{\sum_{n=1}^N (y_n - \hat{y}_n)^2}{N}} \quad (3.15)$$

Onde y_n é a resposta observada para a amostra n , \hat{y}_n a previsão correspondente, e N o número de amostras do conjunto de teste.

3.5.2 Coeficiente de Determinação - R^2

O coeficiente de determinação é uma medida de estatística que calcula a quantidade da variação total da variável de saída que é explicada pela variação das variáveis de entrada, ou seja observa quão próximos os dados estão da linha de regressão ajustada. R^2 é calculada através da Equação 3.16.

$$R^2 = \frac{\sum_{n=1}^N (\hat{y}_n - \bar{y})^2}{\sum_{n=1}^N (y_n - \bar{y})^2} = \frac{\sum_{n=1}^N (y_n - \bar{y})^2 - \sum_{n=1}^N (y_n - \hat{y}_n)^2}{\sum_{n=1}^N (y_n - \bar{y})^2} = 1 - \frac{\sum_{n=1}^N (y_n - \hat{y}_n)^2}{\sum_{n=1}^N (y_n - \bar{y})^2} \quad (3.16)$$

Onde y_n é a resposta observada da amostra n , \bar{y} a média de y , \hat{y}_n a resposta estimada, $\sum_{n=1}^N (y_n - \bar{y})^2$ é a variabilidade observada, $\sum_{n=1}^N (\hat{y}_n - \bar{y})^2$ é a variabilidade explicada pela análise de regressão (pelo modelo) e $\sum_{n=1}^N (y_n - \hat{y}_n)^2$ é a variabilidade que não foi explicada pelo modelo de regressão.

Para os modelos lineares de regressão, o R^2 varia de $[0, 1]$. Quando R^2 é valor nulo, a variável de resposta não consegue ser estimada pelos preditores, por outro lado, quando é igual à unidade, existe uma relação perfeita entre as amostras observadas e as estimadas, e o modelo de regressão tem um excelente desempenho preditivo.

3.6 Comparação dos Modelos

Após uma descrição dos modelos de regressão usados na tese, bem como das métricas usadas para avaliar o desempenho preditivo de cada um, nesta secção descreve-se o processo de comparação entre os vários modelos implementados.

A metodologia de comparação usada permite uma estimação robusta do(s) hiper-parâmetro(s) de cada um dos modelos de regressão para os diferentes conjuntos de dados apresentados na Secção 4.2. De notar que, para que a comparação preditiva seja feita de forma justa e robusta, os conjuntos de dados de treino e de teste são exatamente os mesmos para cada um dos modelos de regressão. No Algoritmo 1 encontra-se representado o algoritmo computacional usado. O algoritmo começa pelo ciclo externo, na qual se escolhe o número de repetições das operações internas, n_{roi} , com o objetivo de aumentar a robustez da comparação (linha 1). A primeira operação interna é uma divisão aleatória do conjunto de dados, num conjunto de dados de treino e num conjunto de dados de teste. O *ratio* estabelecido foi de 80% dos dados para o treino e 20% dos dados para o conjunto de teste (linha 2). De seguida, efetua-se a normalização dos dados, na qual se usou maioritariamente a normalização *z-score* (linha 3). Após as primeiras operações internas, o conjunto de dados de treino é usado para sintonizar (otimizar a seleção) os hiper-parâmetros de cada um dos modelos de regressão, através de uma validação cruzada de 5-folds (linha 5). A 5-F CV, divide o conjunto de treino em cinco folds, em que quatro deles são usados para treinar o modelo, e o fold restante é usado para realizar a validação cruzada. Este processo repete-se cinco vezes, sempre com folds com dados diferentes, uma vez que cada fold é usado uma única vez no processo de validação. Para cada vez que o processo se repete (5-F CV), o(s) hiper-parâmetro(s) do modelo de regressão tomam valores diferentes, obtendo-se diferentes valores de RMSE para o fold de validação. O(s) hiper-parâmetro(s) que levam ao menor valor de RMSE, são guardados e adotados no processo de desenvolvimento do modelo (linha 6). De seguida, com o modelo estimado faz-se a previsão das observações no conjunto de dados de teste (linha 7), e calcula-se os erros de previsão com as métricas de desempenho, $RMSE^{test}$ e R_{test}^2 , apresentadas nas Equações 3.15 e 3.16, respetivamente (linha 8). Uma vez que são repetidas estas operações internas, n_{roi} vezes, os valores das métricas obtidas na linha 8 do Algoritmo 1, vão sendo guardados numa tabela, e no fim do ciclo externo, são calculadas e apresentadas as médias de cada uma (linha 9 e 10). O modelo de regressão que apresentar o menor valor de média do $RMSE^{test}$ e o maior valor de média do R_{test}^2 , considera-se que foi o modelo que apresentou melhor capacidade preditiva para o conjunto de dados.

Algoritmo 1 Pseudo-código da comparação preditiva dos modelos de regressão

- 1: **for** $j = 1 : n_{roi}$ **do**
 - 2: Divisão aleatória do conjunto de dados em conjunto de treino (80%) e teste (20%);
 - 3: Normalização dos conjuntos de dados;
 - 4: **for** $m = 1, 2, \dots, M$ **do**
 - 5: O conjunto de dados de treino é usado para sintonizar o(s) hiper-parâmetro(s) usando 5-fold cross-validation;
 - 6: Estimação do modelo com o conjunto de dados de treino e o(s) hiperparâmetro(s) selecionados;
 - 7: Previsão de observações no conjunto de dados de teste;
 - 8: Cálculo do $RMSE_{j,m}^{test}$ e do $R_{j,m}^{2test}$, onde j representa o número da operação interna, e m o índice do método de regressão no momento.
 - 9: Cálculo da média do RMSE e R^2 ;
 - 10: Apresentação dos resultados de ambas métricas de desempenho.
-

4

Resultados e Discussão

Conteúdo

4.1	Introdução	36
4.2	Conjunto de Dados para Modelação do Sensor Virtual	36
4.3	Hiper-parâmetros dos Modelos de Regressão	37
4.4	Resultados	37

4.1 Introdução

No desenvolvimento desta dissertação foram efetuados testes ao nível do desempenho preditivo para os métodos de regressão com seis conjuntos de dados disponíveis em repositórios públicos, apresentados na Secção 4.2. Em todos os conjuntos de dados, dividiu-se de forma aleatória o conjunto de dados em 80% dos dados para treino e 20% para teste. Foi usada a estratégia de validação cruzada de 5-*folds*, explicada no final do Capítulo 3, na Secção 3.6.

A Secção 4.3 descreve como foi feita a seleção dos hiper-parâmetros para cada um dos modelos de regressão. A partir deste ponto, cada um dos conjuntos de dados será brevemente apresentado, mostrando-se parâmetros estatísticos relativos à sua resposta, bem com as variáveis de entrada e amostras disponíveis para o seu processo de previsão. De realçar que, apesar da fase de seleção de variáveis ser muito importante no desenvolvimento de um sensor virtual, o foco principal foi a de seleção, treino e validação do modelo de regressão: apesar de se implementar e testar um modelo que realiza seleção de variáveis, como o *Stepwise Regression*, os métodos de seleção de variáveis mais usados em aplicações de sensores virtuais não foram abordados.

Ao longo da Secção 4.4 serão apresentados todos os resultados, tanto de treino como de teste, obtidos para cada um dos modelos de regressão.

4.2 Conjunto de Dados para Modelação do Sensor Virtual

Nesta secção são apresentados e sumarizados os conjuntos de dados públicos que se usaram para avaliar e validar o sensor virtual [57, 58]:

1. **Box-Jenkins gas furnace process data:** o conjunto de dados do forno a gás Box-Jenkins foi recolhido a partir de um processo de combustão de uma mistura metano-ar. O conjunto de dados é constituído por 290 amostras $[y(t), u(t)]$. A variável de entrada $u(t)$ é o fluxo de gás dentro do forno e a variável de saída $y(t)$, que se pretende prever, é a concentração de dióxido de carbono (CO_2). Ambas as variáveis são amostradas a cada 9 segundos [27].
2. **Friedman:** este conjunto de dados foi usado inicialmente pelo autor do artigo [59]. A função Friedman original contém cinco variáveis independentes (x_1, x_2, x_3, x_4, x_5) e o valor da variável de resposta y é obtido através da Equação 4.1.

$$y = 10\sin(\pi x_1 x_2) + 20(x_3 - 0.5)^2 + 10x_4 + 5x_5 + \varepsilon \quad (4.1)$$

Adicionaram-se mais cinco variáveis de entrada ao conjunto de dados, no entanto, não têm influência na variável de saída. Todo o espaço de entrada está distribuído entre $[0, 1]$. Este conjunto de dados inclui 40768 observações.

3. **Boston Housing Price:** este conjunto de dados contém informação recolhida pelo *U.S Census Service* relativamente a casas na área de Boston, Massachusetts. É muito frequente ser usado na literatura por métodos de referência. É constituído por 13 variáveis independentes, maioritariamente sócio-económicas, e o objetivo é prever o preço médio de casas em Boston. O conjunto de dados é pequeno, composto por 506 amostras. Estes dados foram originalmente publicados no artigo [60].
4. **Debutanizer column:** conjunto de dados relativo a um processo numa refinaria. O objetivo é prever a concentração de butano numa coluna debutanizadora usando 7 variáveis de entrada. Estão disponíveis 2394 amostras. A medição da concentração de butano na refinaria é feita com 45 minutos de atraso [14].
5. **Coarse Return in a Cement Industry:** o objetivo é prever o retorno grosseiro das partículas num processo de moagem de cimento, uma vez que é um dos principais parâmetros que representa a qualidade do produto final. O conjunto de dados é composto por 6 variáveis de entrada e a variável de saída (*coarse return*), recolhidas a cada minuto durante 108h, correspondendo a cerca de 6500 amostras. A previsão da variável de saída é calculada para os 1/5/10/15 minutos seguintes tal como fez o artigo [61].
6. **pH neutralization process:** representa dados de simulação de um processo de neutralização de *pH*, num tanque de agitação. O objetivo é prever o *pH* da solução no tanque. O conjunto de dados é composto por 2 variáveis de entrada [$u_1(t), u_2(t)$], o fluxo de uma solução ácida em litros e o fluxo de uma solução básica em litros, respetivamente, contendo 2001 amostras. As variáveis são recolhidas com um intervalo de amostragem de 10 segundos.

4.3 Hiper-parâmetros dos Modelos de Regressão

A maioria dos modelos de regressão abordados necessitam de uma otimização na seleção dos seus hiper-parâmetros, para que se obtenham os melhores resultados preditivos em cada um dos conjuntos de dados utilizados. O Algoritmo 1, apresentado na Secção 3.6, descreve o processo de calibração dos hiper-parâmetros para cada modelo de regressão. Na Tabela 4.1 são apresentados os intervalos e listas de valores usados para cada hiper-parâmetro e a estratégia de seleção usada por cada modelo de regressão, de forma a que se conseguisse garantir que o valor ideal, para cada caso, estivesse dentro dessa gama de valores.

4.4 Resultados

Antes de apresentar os resultados para cada um dos conjuntos de dados utilizados, na Tabela 4.2 é apresentado o nome pelo qual cada um dos conjuntos de dados vai ser tratado, tal como o seu número de amostras, N , e o seu número de variáveis de entrada, m . De seguida, vão ser

4. Resultados e Discussão

Tabela 4.1 Seleção dos Hiper-parâmetros usados em cada um dos modelos de regressão durante a fase de treinamento do modelo.

Modelo Regressão	Hiperparâmetro(s)	Valores	Estratégia de Seleção
MLR	-	-	-
SR	threshold-in	0.05	-
	threshold-out	0.10	
RR	α	0	5-fold CV
	λ	0,0001; 0,001; 0,01; 1; 5; 10	
LASSO	α	1	5-fold CV
	λ	0,0001; 0,001; 0,01; 1; 5; 10	
EN	α	0:1 (0,1)	5-fold CV
	λ	0,0001; 0,001; 0,01; 1; 5; 10	
SVR-RBF	C	1; 10; 100; 1000; 10000	5-fold CV
	ϵ_{rbf}	0,001; 0,005; 0,01; 0,05; 0,1	
	γ	0,0001; 0,001; 0,01; 0,1; 0,2; 0,5; 0,6; 0,9	
SVR-Linear	C	1; 10; 100; 1000; 10000	5-fold CV
	ϵ_{linear}	0,001; 0,005; 0,01; 0,05; 0,1	

apresentados os resultados das métricas de desempenho de cada um dos modelos de regressão, para cada um dos conjuntos dos dados referidos.

Tabela 4.2 Visão geral do Conjunto de Dados utilizados.

Conjunto de Dados	Nomenclatura	N	m
Box-Jenkins Gas Furnace Process Data	Box-Jenkins	296	10
Friedman	Friedman	1000	10
Boston Housing Price	Boston Housing	506	13
Debutanizer Column	Debutanizer Column	2394	7
Course Return in a Cement Industry	Cement Industry	6570	6
pH Neutralization Process	pH	2001	2

4.4.1 Resultados 'Box-Jenkins'

O Box-Jenkins, como descrito na Secção 4.2, refere-se a dados recolhidos a partir de um processo de combustão de uma mistura metano-ar. Originalmente, é constituído por 296 amostras, cada uma composta pela variável de entrada $u(t)$, fluxo de gás dentro do forno, e pela saída $y(t)$, a concentração de dióxido de carbono que sai do forno, a variável de resposta que se quer prever. Na Tabela 4.3 é sumarizada a variável de resposta CO_2 , apresentado-se o número de amostras disponíveis e alguns parâmetros estatísticos.

Tabela 4.3 Número de amostras de CO_2 e parâmetros estatísticos.

Variável de Resposta	Número de Amostras	Mínimo	Máximo	Média	Desvio Padrão
CO2	296	-2.716	2.834	-0.059	1.083

Foram efetuados atrasos temporais ao nível das duas variáveis originais, $[y(t), u(t)]$, na qual

o conjunto de variáveis de entrada passou a ser o seguinte: $[y(t-1), y(t-2), y(t-3), y(t-4), u(t-1), u(t-2), u(t-13), u(t-4), u(t-5), u(t-6)]$. As variáveis de entrada que se usaram, foram as mesmas que o método proposto em [27]. Assim, o conjunto resultante de variáveis de entrada foi o seguinte: $[y(t-1), y(t-2), y(t-3), y(t-4), u(t-1), u(t-5), u(t-6)]$. Ao conjunto de dados resultante, realizou-se um pré-processamento, de forma a que se encontrasse possíveis valores em falta e/ou *outliers*. Na Tabela 4.4 são apresentados os valores das médias das métricas de desempenho ($RMSE$ e R^2), nas condições de treino e teste. obtidos durante o processo de validação cruzada apresentado no Algoritmo 1.

Tabela 4.4 Média das métricas de desempenho no Box-Jenkins.

	Modelo de Regressão	Treino		Teste	
		$RMSE^{treino}$	$R^{2treino}$	$RMSE^{teste}$	R^{2teste}
Box-Jenkins	MLR	0.129	0.985	0.171	0.967
	SR	0.135	0.984	0.151	0.973
	RR	0.127	0.984	0.139	0.981
	LASSO	0.128	0.985	0.141	0.980
	EN	0.128	0.985	0.140	0.979
	SVR(rbf)	0.124	0.986	0.129	0.982
	SVR(Linear)	0.120	0.988	0.126	0.983

A maioria dos modelos de regressão apresentados na Tabela 4.4, apresentaram bons resultados, em relação ao desempenho de previsão da variável de resposta, CO_2 . Os modelos SVR (linear), SVR (rbf) e RR foram os que tiveram uma certa vantagem perante as outras abordagens. Nas Figuras 4.1–4.3, são apresentados os gráficos, tanto para o treino como para o teste, entre os valores observados e as suas previsões, para os modelos mencionados anteriormente. Os modelos de regressão dos restantes modelos estudados, podem ser acedidos no Anexo A.1.

4. Resultados e Discussão

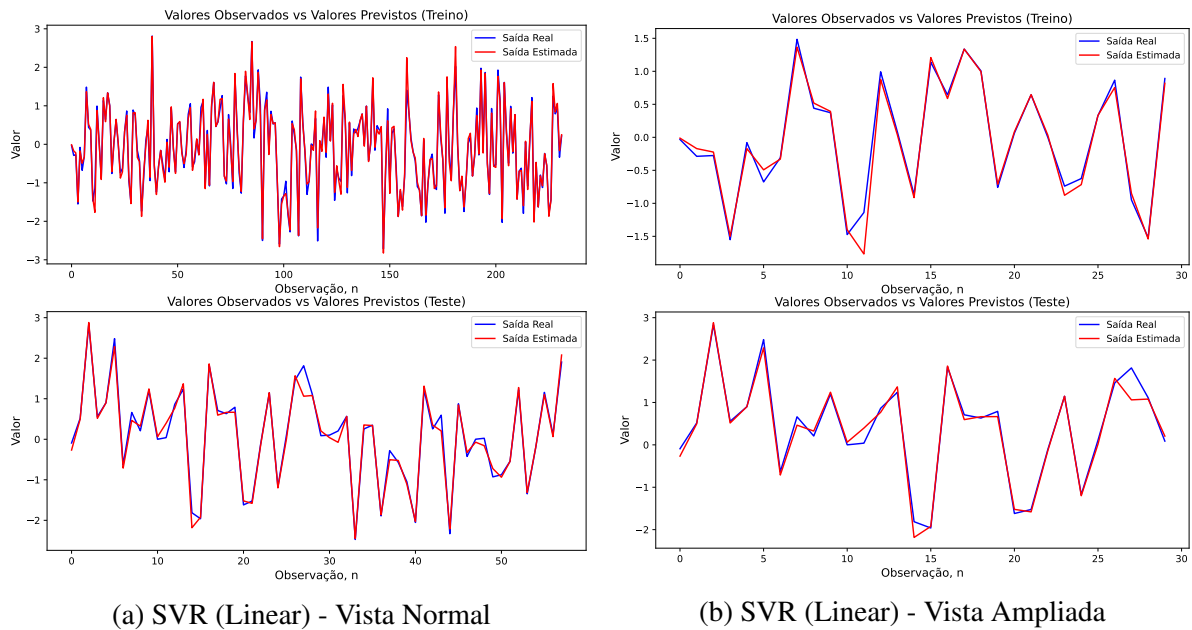


Figura 4.1: Resultados de treino e teste do SVR (Linear), com vista normal e vista ampliada (Box-Jenkins).

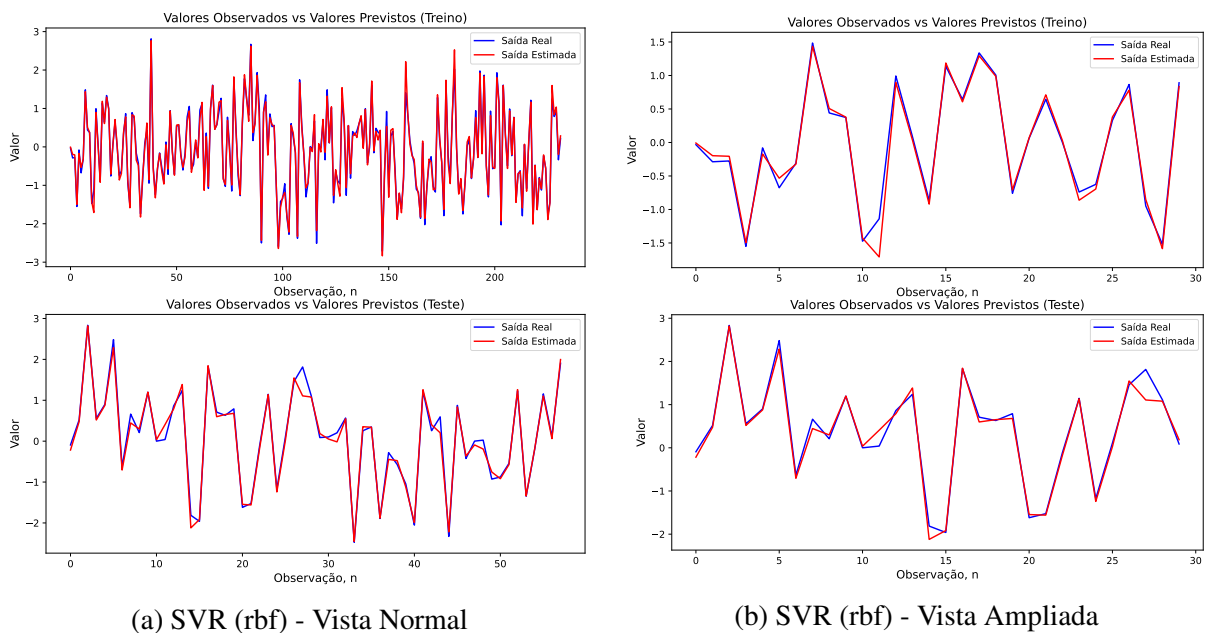


Figura 4.2: Resultados de treino e teste do SVR (rbf), com vista normal e vista ampliada (Box-Jenkins).

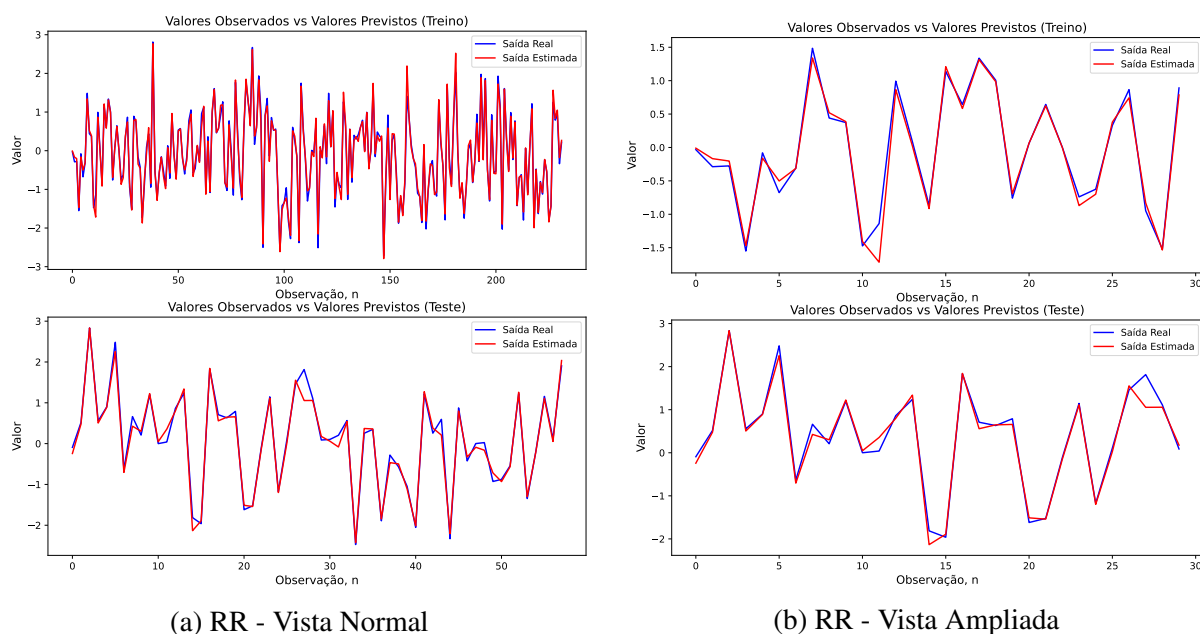


Figura 4.3: Resultados de treino e teste do RR, com vista normal e vista ampliada (Box-Jenkins).

4.4.2 Resultados 'Friedman'

O conjunto de dados 'Friedman', como descrito na Secção 4.2, refere-se a dados obtidos a partir da Equação 4.1. Na Tabela 4.5 é resumida a variável de resposta.

Tabela 4.5 Número de amostras de Output e parâmetros estatísticos.

Variável de Resposta	Número de Amostras	Mínimo	Máximo	Média	Desvio Padrão
Output	1000	0.91	26.93	14.46	5.14

Na Tabela 4.6 apresentam-se os valores das médias das métricas de desempenho, obtidos pelo processo de validação cruzada de 5-*folds*.

Tabela 4.6 Média das métricas de desempenho no Friedman.

Modelo de Regressão	Treino		Teste	
	$RMSE^{treino}$	$R^{2^{treino}}$	$RMSE^{teste}$	$R^{2^{teste}}$
MLR	2.631	0.724	2.641	0.709
SR	2.559	0.750	2.601	0.718
RR	2.545	0.752	2.768	0.721
LASSO	2.545	0.752	2.765	0.722
EN	2.546	0.752	2.765	0.722
SVR(rbf)	1.311	0.974	1.625	0.904
SVR(Linear)	2.597	0.742	2.774	0.720

Analisando a Tabela 4.6, conclui-se que todos os modelos de regressão apresentaram um

4. Resultados e Discussão

desempenho razoável e boa consistência nos resultados para os dados de treino e de teste. No entanto, o SVR (rbf), foi claramente o modelo que melhor desempenho preditivo apresentou ao longo do processo de modelagem dos dados.

Nas Figuras 4.4–4.6 são apresentados, graficamente, os resultados de previsão do SVR (rbf), MLR e SR. Apesar de não se ter aplicado nenhum método de seleção de variáveis aos modelos de regressão utilizados, o SR, que aplica o método dos mínimos quadrados para obter os coeficientes de regressão que explicam a variável de resposta, selecionou a primeira, a segunda, a quarta e a quinta variáveis de entrada para a regressão dos dados. De salientar que todas as variáveis de entrada selecionadas pelo SR, não correspondem às variáveis que não influenciam a variável de resposta (x_6 , x_7 , x_8 , x_9 e x_{10}), tal como foi referido na descrição do 'Friedman' (Secção 4.2). Os modelos de regressão que não foram mostrados graficamente, podem ser acessados no Anexo A.2.

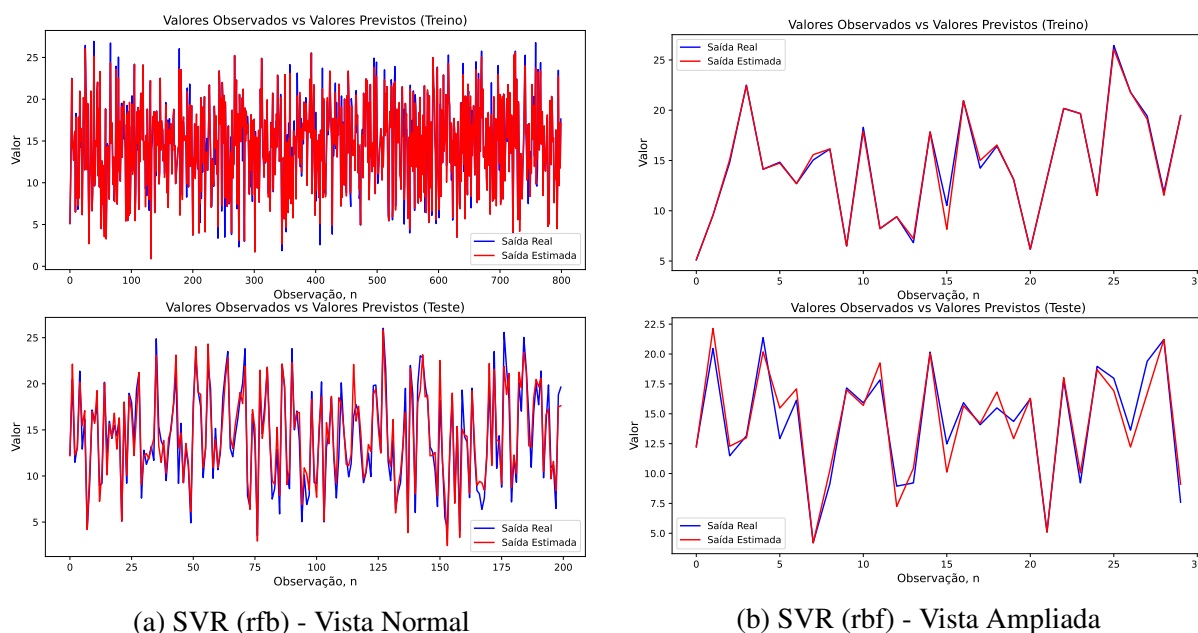


Figura 4.4: Resultados de treino e teste do SVR (rbf), com vista normal e vista ampliada (Friedman).

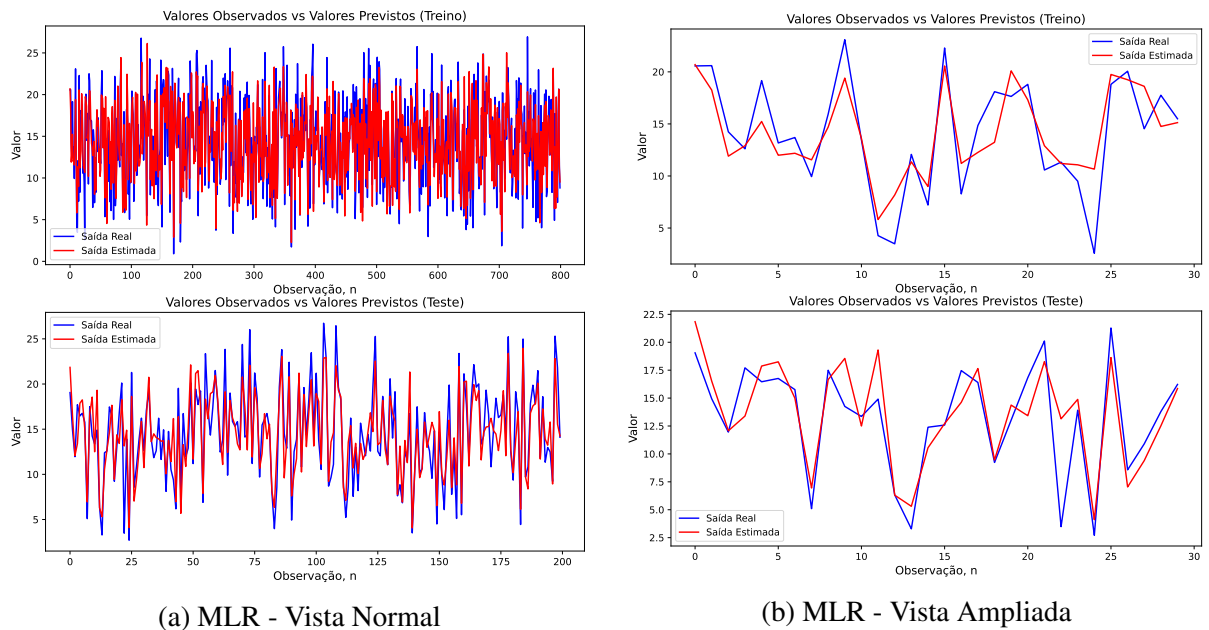


Figura 4.5: Resultados de treino e teste do MLR, com vista normal e vista ampliada (Friedman).

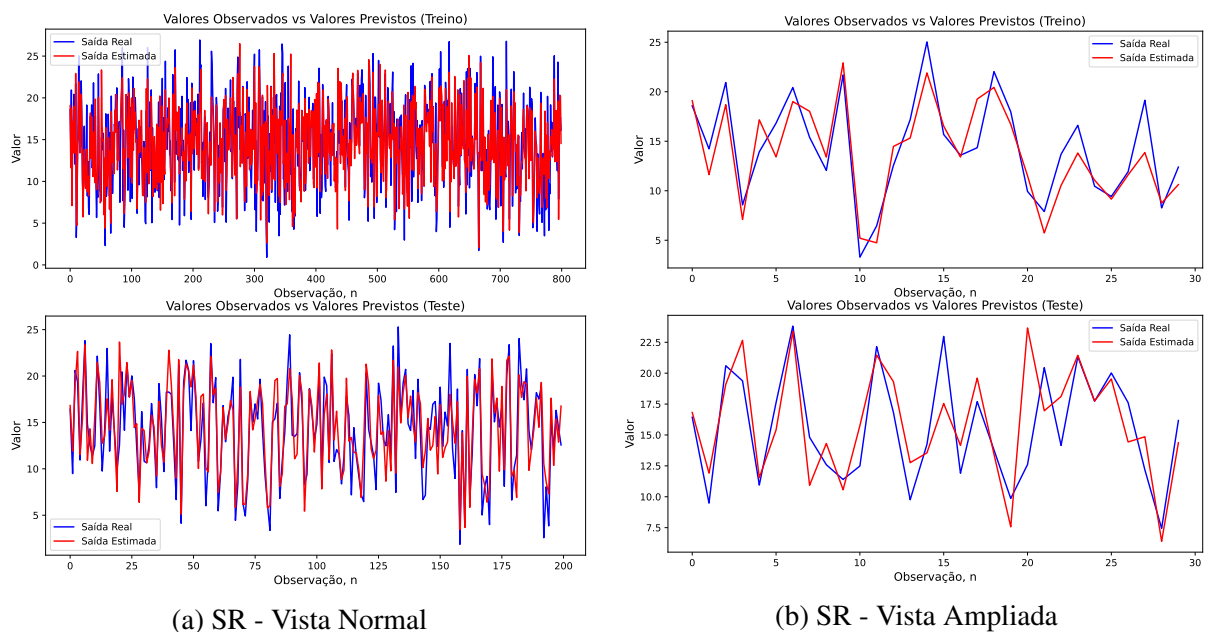


Figura 4.6: Resultados de treino e teste do SR, com vista normal e vista ampliada (Friedman).

4.4.3 Resultados 'Boston Housing'

O conjunto de dados 'Boston Housing', descrito na Secção 4.2, reúne informações de 13 variáveis que tentam prever o valor médio de casas em Boston. Na Tabela 4.7 é resumida a variável de resposta.

4. Resultados e Discussão

Tabela 4.7 Número de amostras de valores médios de casas em Boston e parâmetros estatísticos.

Variável de Resposta	Número de Amostras	Mínimo	Máximo	Média	Desvio Padrão
Valor Médio	506	5	50	22.53	9.20

Inicialmente, ao conjunto de dados, realizou-se um pré-processamento, de forma a encontrar possíveis valores em falta e/ou *outliers*. A Tabela 4.8 fornece os resultados do desempenho preditivo de cada um dos modelos de regressão.

Tabela 4.8 Média das métricas de desempenho no Boston Housing.

Modelo de Regressão	Treino		Teste	
	$RMSE^{treino}$	$R^{2treino}$	$RMSE^{teste}$	R^{2teste}
MLR	4.624	0.741	4.814	0.722
SR	4.173	0.765	4.724	0.733
RR	4.522	0.751	5.409	0.686
LASSO	4.513	0.751	5.372	0.691
EN	4.514	0.752	5.383	0.689
SVR(rbf)	2.548	0.921	3.102	0.896
SVR(Linear)	4.757	0.724	5.753	0.645

A maioria dos modelos, como se pode ver pela Tabela 4.8, apresentam um desempenho aceitável em termos de precisão de previsão. Os resultados dão uma certa vantagem ao SVR (rbf), ao SR e ao MLR. A seleção de variáveis realizada pelo SR melhorou um pouco o seu desempenho preditivo em relação ao MLR. Das treze variáveis disponíveis, o SR selecionou todas exceto as variáveis número 3 e 7. De seguida, serão apresentados os gráficos de comparação entre o valor real e o valor previsto dos três melhores modelos 4.7–4.9. Os modelos de regressão que não foram apresentados nesta secção, podem ser acedidos ao Anexo A.3.

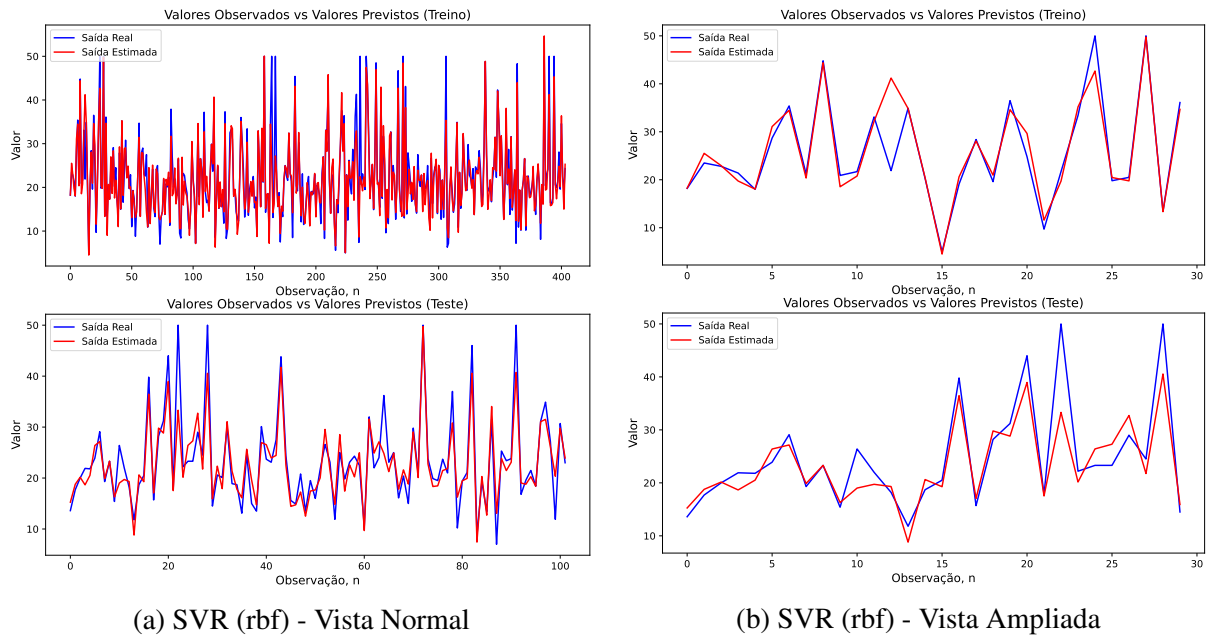


Figura 4.7: Resultados de treino e teste do SVR (rbf), com vista normal e vista ampliada (Boston Housing).

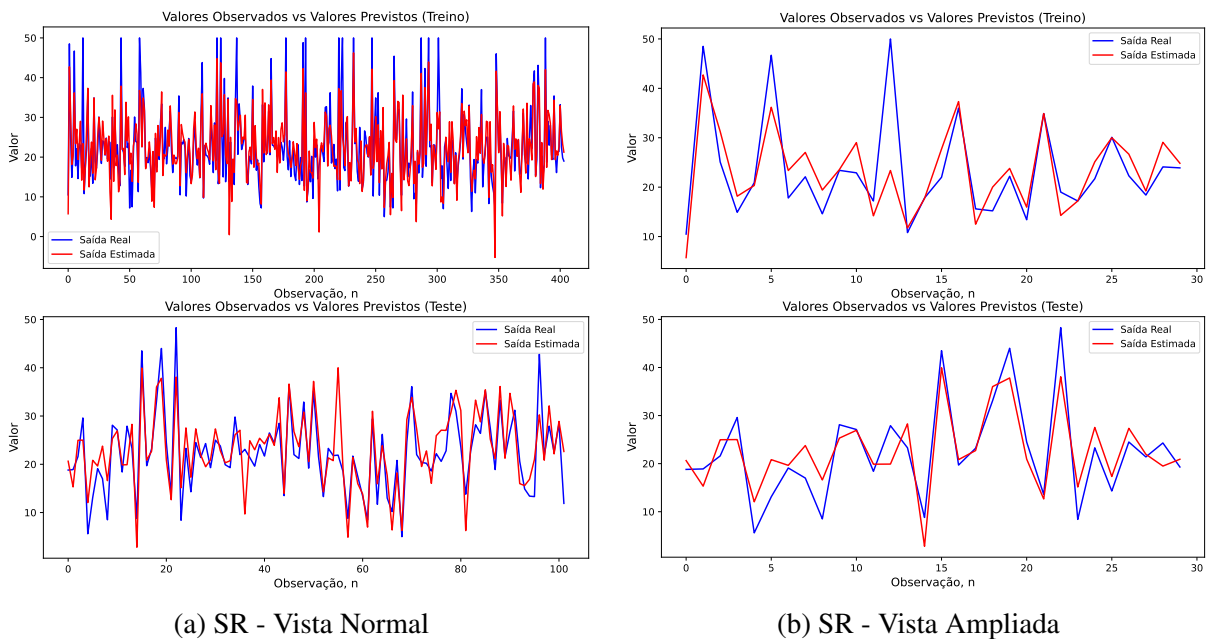


Figura 4.8: Resultados de treino e teste do SR, com vista normal e vista ampliada (Boston Housing).

4. Resultados e Discussão

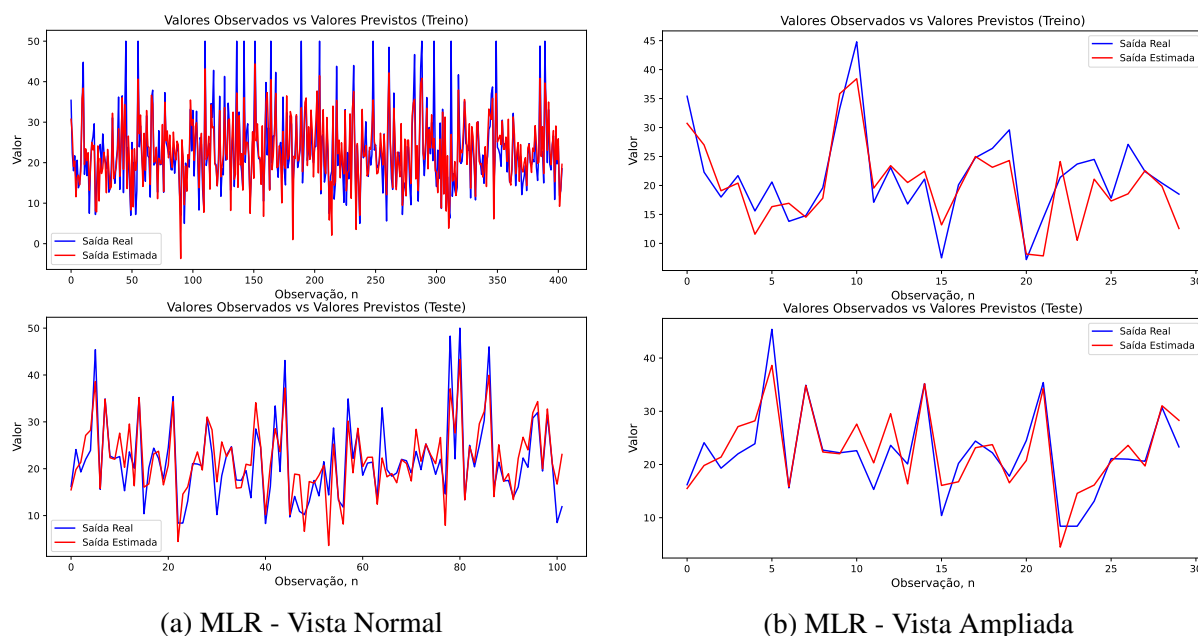


Figura 4.9: Resultados de treino e teste do MLR, com vista normal e vista ampliada (Boston Housing).

4.4.4 Resultados 'Debutanizer Column'

Este caso de estudo consiste na previsão da concentração de butano no fluxo do fundo de uma coluna debutanizadora. A 'Debutanizer Column' é composta por sete variáveis de entrada correspondentes a temperaturas, pressões e fluxos e a variável de saída. Na Tabela 4.9 apresenta-se os dados disponíveis e dados estatísticos relativamente à concentração de butano. Os resultados do desempenho preditivo obtido por cada um dos modelos de regressão são apresentados na tabela 4.10.

Tabela 4.9 Número de amostras de Concentração de Butano e parâmetros estatísticos.

Variável de Resposta	Número de Amostras	Mínimo	Máximo	Média	Desvio Padrão
Concentração de Butano	2394	0	1	0.27	0.16

Inicialmente, sobre o conjunto de dados, realizou-se um pré-processamento, de forma a encontrar possíveis valores em falta e/ou outliers. A Tabela 4.10 fornece os resultados do desempenho preditivo de cada um dos modelos de regressão.

Tabela 4.10 Média das métricas de desempenho no Debutanizer.

	Modelo de Regressão	Treino		Teste	
		$RMSE^{treino}$	$R^{2treino}$	$RMSE^{teste}$	R^{2teste}
Debutanizer Column	MLR	0.143	0.204	0.133	0.256
	SR	0.139	0.220	0.149	0.168
	RR	0.141	0.219	0.141	0.182
	LASSO	0.147	0.218	0.141	0.177
	EN	0.141	0.214	0.141	0.178
	SVR(rbf)	0.047	0.916	0.072	0.788
	SVR(Linear)	0.142	0.212	0.142	0.175

Os resultados obtidos realçam a grande capacidade preditiva do SVR (RBF) que consegue obter um valor mínimo de $RMSE^{teste}$ e ainda consegue obter um valor favorável de R^{2teste} . Os resultados do $RMSE^{teste}$ para os restantes modelos de regressão foram muito bons, no entanto, tendo em conta os valores obtidos para o R^{2teste} , conclui-se que as variáveis de entrada não conseguem explicar a variável de saída em condições. Os valores mínimos de R^{2teste} obtidos, podem ser devido ao facto do "Debutanizer Column" ser um conjunto de dados em que se observa não-linearidade entre as variáveis de entrada e a variável de saída: de facto o único modelo com resultados bastante positivos foi o SVR (RBF), correspondendo a um modelo não-linear. Na Figura 4.10 apresenta-se o gráfico de comparação entre o valor real da concentração de butano e o seu valor previsto. Os modelos de regressão que não foram apresentados nesta secção, podem ser acedidos ao Anexo A.4.

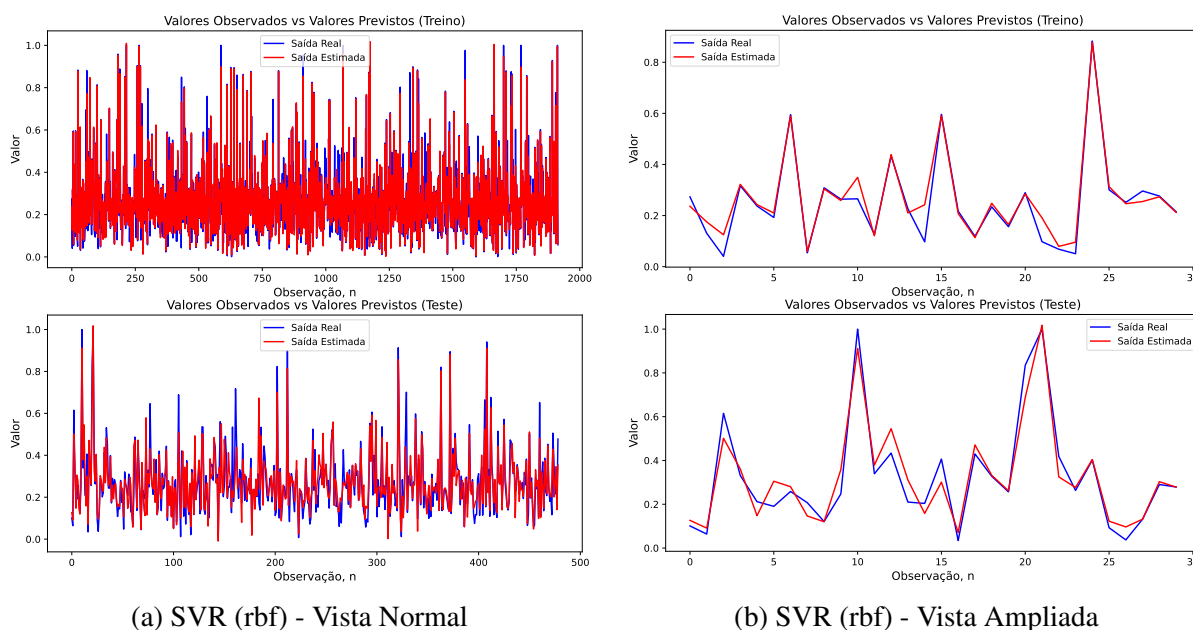


Figura 4.10: Resultados de treino e teste do SVR (rbf), com vista normal e vista ampliada (Debutanizer Column).

4. Resultados e Discussão

4.4.5 Resultados 'Cement Industry'

Este caso de estudo corresponde a dados de uma indústria de cimento, como explicado na Secção 4.2. A variável de resposta que se quer prever é apresentada na Tabela 4.11.

Tabela 4.11 Número de amostras de 'Coarse Return' e parâmetros estatísticos.

Variável de Resposta	Número de Amostras	Mínimo	Máximo	Média	Desvio Padrão
Coarse Return	6570	13.84	200	89.95	40.20

Os modelos de regressão foram aplicados a um conjunto de dados de teste apresentado no artigo [62]. O objetivo é prever o *Coarse Return* para os 1/5/10/15 minutos seguintes. Começou-se por realizar atrasos temporais no conjunto de dados. Os resultados para os diferentes atrasos temporais são mostrados na Tabela 4.12. Graficamente, serão apenas mostrados para o atraso de 15 minutos, tal como fez o artigo [62].

Tabela 4.12 Média das métricas de desempenho para os diferentes níveis de atrasos temporais

Modelo	1min		5min		10min		15min	
	$RMSE^{teste}$	$R^{2^{teste}}$	$RMSE^{teste}$	$R^{2^{teste}}$	$RMSE^{teste}$	$R^{2^{teste}}$	$RMSE^{teste}$	$R^{2^{teste}}$
MLR	22.528	0.683	23.110	0.673	24.140	0.604	26.120	0.598
SR	22.850	0.681	22.851	0.678	23.378	0.655	25.467	0.612
RR	22.093	0.713	22.264	0.702	22.854	0.645	23.150	0.628
LASSO	22.090	0.730	22.262	0.704	23.451	0.662	24.244	0.686
EN	22.095	0.694	22.263	0.674	22.910	0.654	23.654	0.640
SVR(rbf)	12.507	0.908	15.137	0.862	16.168	0.835	18.068	0.795
SVR(linear)	22.20	0.710	22.393	0.698	23.640	0.685	24.265	0.664

Pela Tabela 4.12, conclui-se que à medida que aumentamos o atraso temporal de previsão da *coarse return*, os erros de previsão acumulam-se, aumentando o valor de métrica $RMSE^{teste}$ e reduzindo a capacidade de explicar a variável de saída através das variáveis de entrada, ou seja, menor é o valor de $R^{2^{teste}}$. Apesar desta característica ser comum tanto nos resultados apresentados, como nos encontrados no artigo [62], os resultados são aceitáveis. Nas Figuras 4.11 e 4.12 são apresentados os gráficos entre o valor real de *coarse return* e a sua previsão, para o SVR (rbf), que demonstrou ser o melhor modelo de regressão aplicado, e para o método de regularização, LASSO.

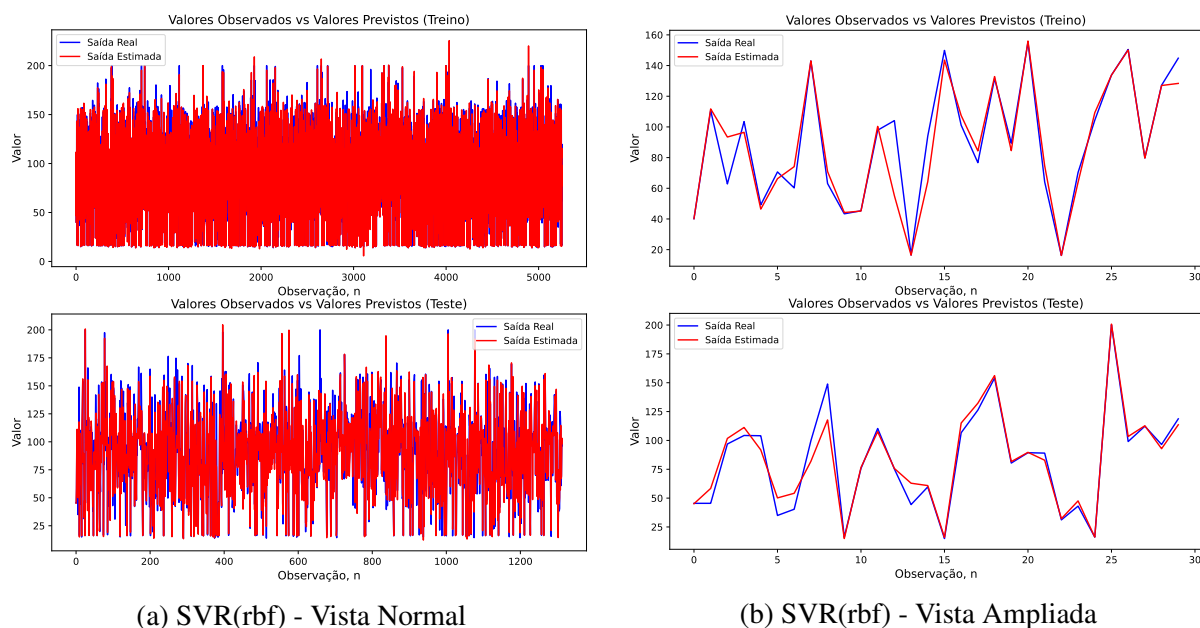


Figura 4.11: Resultados de treino e teste do SVR(rbf), com vista normal e vista ampliada (Cement Industry).

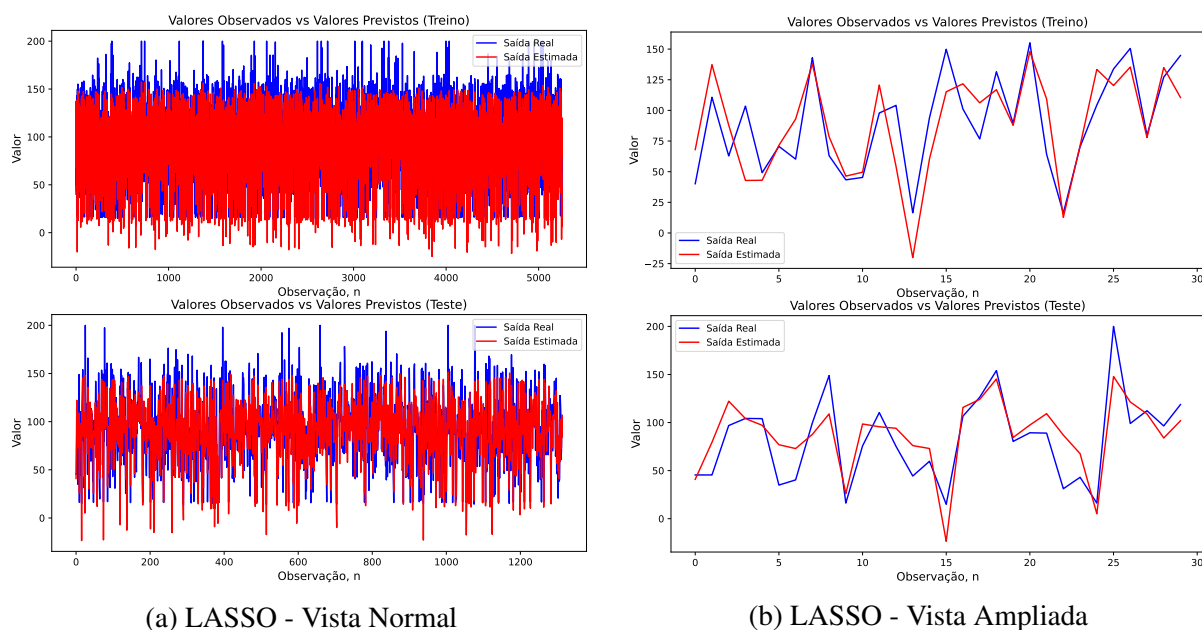


Figura 4.12: Resultados de treino e teste do LASSO, com vista normal e vista ampliada (Cement Industry).

4.4.6 Resultados 'pH'

O último caso de estudo é a previsão dum processo de neutralização do pH num tanque. O sistema em estudo é descrito como de elevada não-linearidade [63]. Na Tabela 4.13 apresentam-se as amostras disponíveis da variável de resposta e alguns parâmetros estatísticos correspon-

4. Resultados e Discussão

Os resultados obtidos por cada um dos modelos de regressão são apresentados na Tabela 4.14.

Tabela 4.13 Número de amostras de pH e parâmetros estatísticos.

Variável de Resposta	Número de Amostras	Mínimo	Máximo	Média	Desvio Padrão
pH	2001	3.69	11.80	9.94	2.62

Tabela 4.14 Média das métricas de desempenho no pH.

Modelo de Regressão	Treino		Teste	
	$RMSE^{treino}$	$R^{2treino}$	$RMSE^{teste}$	R^{2teste}
MLR	2.304	0.238	2.148	0.282
SR	2.277	0.244	2.257	0.256
RR	2.301	0.247	2.159	0.239
LASSO	2.301	0.247	2.160	0.239
EN	2.301	0.246	2.159	0.239
SVR(rbf)	1.971	0.447	1.902	0.410
SVR(Linear)	2.809	-0.122	2.552	-0.062

O desempenho preditivo apresentado pelos modelos foi, em termos gerais, fraco. O modelo de regressão que conseguiu obter um desempenho acima da média foi o SVR (rbf). É mostrado na Figura 4.13 o valor real de pH e as suas previsões.

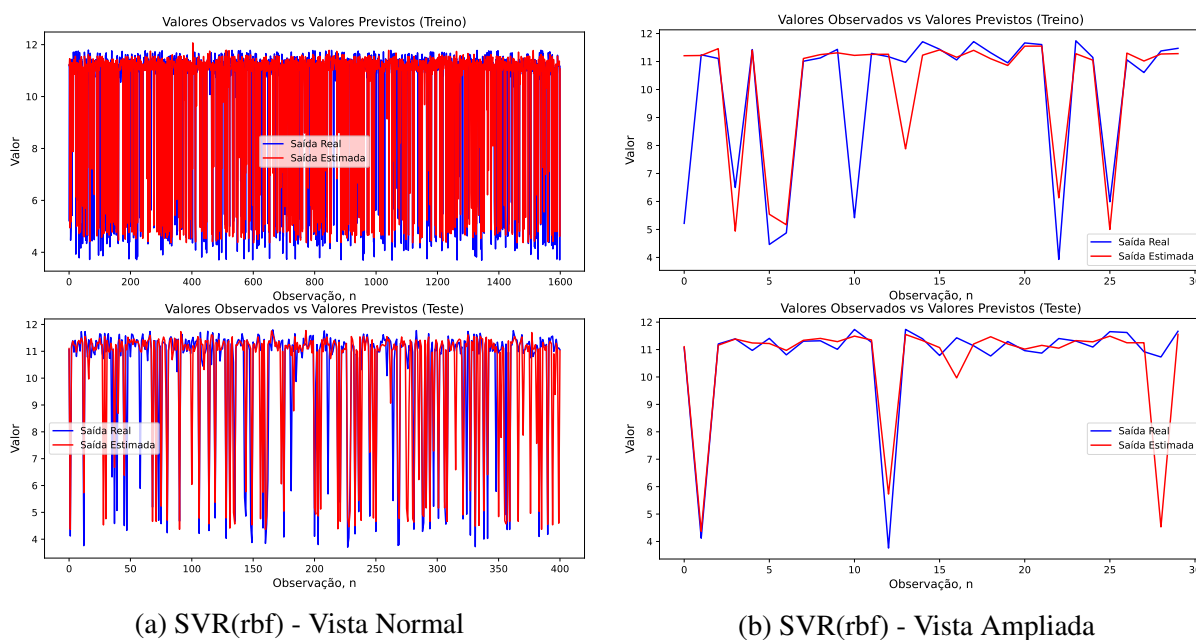


Figura 4.13: Resultados de treino e teste do SVR(rbf), com vista normal e vista ampliada (pH).

5

Conclusões

Conteúdo

5.1 Conclusão	52
5.2 Trabalho Futuro	52

5.1 Conclusão

Esta tese apresentou sete modelos de regressão de aprendizagem supervisionada capazes de realizar regressão sob diversos conjuntos de dados de referência. Qualquer um dos modelos de regressão utilizados, apenas necessita de dados históricos do processo, tanto das variáveis de entrada como da variável de saída, não precisando de conhecimento empírico do processo em que está inserido. Sem o processo de seleção de variáveis realizado, o SVR (rbf) mostrou ser um modelo com um desempenho preditivo muito competitivo em todos os conjuntos de dados usados, exceto no último caso de estudo, na previsão do pH. Os restantes modelos tiveram resultados aceitáveis, não havendo nenhum outro que se superiorizasse significativamente em relação aos outros. O MLR mostrou ser um modelo capaz de realizar regressão sobre conjunto de dados com resultados consistentes, e em certos casos, bastante positivos. O método de seleção de variáveis, SR, viu a sua capacidade preditiva superiorizar-se, em certos casos, à do modelo MLR, o que até é expectável, já que seleciona o conjunto de variáveis que melhor explica a variável de saída e, de seguida, aplicam-nas ao modelo MLR. Os modelos de regressão de regularização obtêm resultados muito parecidos uns com os outros, não se sobressaindo nenhum em relação aos outros. O modelo vetor de suporte com função linear, SVR (linear), no primeiro conjunto de dados (Box-Jenkins) foi o melhor modelo de regressão, e para os restantes conjuntos de dados utilizados, foi sempre um modelo que cumpriu com a sua função, obtendo valores medianos. Em todos os conjuntos de dados que são compostos por poucas variáveis de entrada, os resultados preditivos de cada um dos modelos de regressão foram menos positivos.

Os resultados obtidos por cada um dos modelos de regressão, em todos os conjuntos de dados utilizados, foram importantes para validar e comprovar, mais uma vez, a relevância dos sensores virtuais em aplicações industriais. A capacidade de fornecer previsões de variáveis críticas ao processo é uma verdadeira ferramenta, de uma extrema importância no controlo contínuo do processo, especialmente em cenários que apenas se consigam obter informações da variável de resposta através de análises laboratoriais, levando a perda de tempo e possível perda de informação das dinâmicas do processo em tempo real.

5.2 Trabalho Futuro

As direções futuras desta tese apontam para a continuação do estudo de modelos de regressão aplicados aos sensores virtuais. O aprofundamento do estudo de métodos de seleção de variáveis pode ser um dos pontos em mais destaque, já que poderia elevar o desempenho preditivo de cada um dos modelos de regressão implementados. A combinação de modelos de regressão pode ser outro procedimento de estudo, na medida em que traria maior leque de opções no desenvolvimento do sensor virtual. Seria interessante testar os modelos de regressão utilizados em mais conjuntos de dados industriais, para confirmar os resultados obtidos durante a fase de teste.

Bibliografia

- [1] F. A. A. Souza, R. Araújo, and J. Mendes, “Review of soft sensors methods for regression applications,” *Chemometrics and Intelligent Laboratory Systems*, vol. 152, pp. 69–79, March 2016.
- [2] Y. Jiang, S. Yin, J. Dong, and O. Kaynak, “A review on soft sensors for monitoring, control, and optimization of industrial processes,” *IEEE Sensors Journal*, vol. 21, no. 11, pp. 12 868–12 881, 2021.
- [3] S. Jiao, Y. Gao, J. Feng, T. Lei, and X. Yuan, “Does deep learning always outperform simple linear regression in optical imaging?” *Opt. Express*, vol. 28, no. 3, pp. 3717–3731, February 2020. [Online]. Available: <http://opg.optica.org/oe/abstract.cfm?URI=oe-28-3-3717>
- [4] Q. Sun and Z. Ge, “A survey on deep learning for data-driven soft sensors,” *IEEE Transactions on Industrial Informatics*, vol. 17, no. 9, pp. 5853–5866, 2021.
- [5] R. Rezazadegan and M. Sharifzadeh, *Applications of Artificial Intelligence and Big Data in Industry 4.0 Technologies*, 2022, pp. 121–158.
- [6] L. Fortuna, S. Graziani, A. Rizz, and M. G.Xibilia, *Soft Sensors for Monitoring and Control of Industrial Processes*. Springer, 2007.
- [7] P. Kadlec, B. Gabrys, and S. Strandt, “Data-driven soft sensors in the process industry,” *Comput. Chem. Eng.*, vol. 33, pp. 795–814, 2009. [Online]. Available: <https://doi.org/10.1016/j.compchemeng.2008.12.012>
- [8] F. Souza, R. Araújo, S. Soares, and J. Mendes, “Variable selection based on mutual information for soft sensors applications,” in *Proc. 9th Portuguese Conference on Automatic Control (CONTROLO 2010)*, Coimbra, Portugal, September 8-10 2010, pp. 1–6.
- [9] Khatibisepehr, B. S.and Huang, and S. Khare, “Design of inferential sensors in the process industry: A review of bayesian methods,” *J. Process Control*, vol. 23, pp. 1575–1596, 2013. [Online]. Available: <https://doi.org/10.1016/j.jprocont.2013.05.007>

- [10] F. Souza, J. Mendes, and R. Araújo, “A regularized mixture of linear experts for quality prediction in multimode and multiphase industrial processes,” *Applied Sciences*, vol. 11, no. 5, p. 2040, February 2021.
- [11] Savytskyi, Oleksandr, Tymoshenko, Maksym, Hramm, Oleksandr, and Romanov, Serhii, “Application of soft sensors in the automated process control of different industries,” *E3S Web Conf.*, vol. 166, p. 05003, 2020. [Online]. Available: <https://doi.org/10.1051/e3sconf/202016605003>
- [12] M. Serpas, Y. Chu, and J. Hahn, “Fault detection approach for systems involving soft sensors,” *Journal of Loss Prevention in the Process Industries*, vol. 26, p. 443–452, 05 2013.
- [13] A. K. Pani and H. K. Mohanta, “A survey of data treatment techniques for soft sensor design,” *Chemical Product and Process Modeling*, vol. 6, no. 1, 2011. [Online]. Available: <https://doi.org/10.2202/1934-2659.1536>
- [14] L. Fortuna, S. Graziani, A. Rizzo, and M. G. Xibilia, *Soft Sensors for Monitoring and Control of Industrial Processes (Advances in Industrial Control)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2006.
- [15] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. Springer Series in Statistics, Springer New York, 2001.
- [16] J. Brownlee, “Deep learning with python: Develop deep learning models on theano and tensorflow using keras,” *Machine Learning Mastery: Vermont, Australia*, 2016.
- [17] T. Hastie, R. Tibshirani, and J. Friedman, “The elements of statistical learning: Data mining, inference and prediction,” *Springer Science Business Media: Berlin, Germany*, 2009.
- [18] J. M. Jerez, I. Molina, P. J. García-Laencina, E. Alba, N. Ribelles, and L. F. M. Martín, “Missing data imputation using statistical and machine learning methods in a real breast cancer problem,” *Artificial Intelligence in Medicine*, vol. 50, pp. 105–115, 2010.
- [19] F. Souza, P. Santos, and R. Araújo, “Variable and delay selection using neural networks and mutual information for data-driven soft sensors,” in *Proc. 15th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA 2010)*. Bilbao, Spain: IEEE, September 13-16 2010, pp. 1–8.
- [20] V. Hodge and J. Austin, “A survey of outlier detection methodologies,” *Artificial Intelligence Review*, vol. 22, p. 85–126, 2004. [Online]. Available: <https://doi.org/10.1007/s10462-004-4304-y>

- [21] “The mahalanobis distance,” *Chemometrics and Intelligent Laboratory Systems*, vol. 50, no. 1, pp. 1–18, 2000.
- [22] I. Ben-Gal, O. Maimon, and L. Rockach, “Data mining and knowledge discovery handbook: A complete guide for practitioners and researchers,” *Kluwer Academic Publishers*, 2005.
- [23] F. A. A. Souza, R. Araújo, T. Matias, and J. Mendes, “A multilayer-perceptron based method for variable selection in soft sensor design,” *Journal of Process Control*, vol. 23, no. 10, pp. 1371–1378, November 2013.
- [24] A. Jović, K. Brkić, and N. Bogunović, “A review of feature selection methods with applications,” in *2015 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, 2015, pp. 1200–1205.
- [25] N. Draper and H. Smith, *Applied Regression Analysis, 3rd ed.* Wiley Series in Probability and Statistics. John Wiley & Sons, Inc, New York., 1998.
- [26] A. Souza and R. Araújo, “Online mixture of univariate linear regression models for adaptive soft sensors,” *IEEE Transactions on Industrial Informatics*, vol. 10, pp. 937–945, May 2014.
- [27] J. Mendes, F. Souza, R. Araújo, and N. Gonçalves, “Genetic fuzzy system for data-driven soft sensors design,” *Applied Soft Computing*, p. 3237–3245, October 2012.
- [28] J. Mendes, S. Pinto, R. Araújo, and F. Souza, “Evolutionary fuzzy models for nonlinear identification,” in *17th IEEE Conference on Emerging Technologies Factory Automation (ETFA 2012)*, 2012, pp. 1–8.
- [29] S. J. Qi, “Recursive pls algorithms for adaptive data modeling,” *Computers Chemical Engineering*, vol. 22, p. 503–514, 1998.
- [30] J. Ding, H. Shi, D. Jiang, and X. Rong, “Prediction of coal mine gas concentration based on partial least squares regression,” in *2019 Chinese Automation Congress (CAC)*, 2019, pp. 5243–5246.
- [31] D. Aguado, G. Noriega-Hevia, J. Ferrer, A. Seco, and J. Serralta, “Pls-based soft-sensor to predict ammonium concentration evolution in hollow fibre membrane contactors for nitrogen recovery,” *Journal of Water Process Engineering*, vol. 47, p. 102735, 2022. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2214714422001799>

Bibliografia

- [32] A. E. Hoerl and R. W. Kennard, "Ridge regression: Biased estimation for nonorthogonal problems," *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970.
- [33] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996. [Online]. Available: <http://www.jstor.org/stable/2346178>
- [34] E. Ramentol, T. Olsson, and S. Barua, "Machine learning models for industrial applications," in *AI and Learning Systems*, K. Kyprianidis and E. Dahlquist, Eds. Rijeka: IntechOpen, 2021, ch. 7. [Online]. Available: <https://doi.org/10.5772/intechopen.93043>
- [35] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Networks*, vol. 61, pp. 85–117, 2015.
- [36] G. Palm, "Warren mcculloch and walter pitts: A logical calculus of the ideas immanent in nervous activity," in *Brain Theory*, G. Palm and A. Aertsen, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1986, pp. 229–230.
- [37] D. E. Rumelhart, J. L. McClelland, and C. PDP Research Group, Eds., *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Vol. 2: Psychological and Biological Models*. Cambridge, MA, USA: MIT Press, 1986.
- [38] A. Jain, J. Mao, and K. Mohiuddin, "Artificial neural networks: a tutorial," *Computer*, vol. 29, no. 3, pp. 31–44, 1996.
- [39] Y.-S. Park and S. Lek, "Chapter 7 - artificial neural networks: Multilayer perceptron for ecological modeling," in *Ecological Model Types*, ser. Developments in Environmental Modelling, S. E. Jørgensen, Ed. Elsevier, 2016, vol. 28, pp. 123–140.
- [40] S. HAYKIN, *Neural Networks: A Guided Tour*, December 2000, pp. 71–80.
- [41] Y. Liu, X. Yao, and T. Higuchi, "Evolutionary ensembles with negative correlation learning," *IEEE Transactions on Evolutionary Computation*, vol. 4, no. 4, pp. 380–387, 2000.
- [42] T. Matias, F. Souza, R. Araújo, S. Rastegar, and J. Mendes, "Adaptive identification and predictive control using an improved on-line sequential extreme learning machine," in *Proc. of The 40th Annual Conference of the IEEE Industrial Electronics Society (IECON 2014)*. Dallas, TX, USA: IEEE, October 29 - November 1 2014, pp. 58–64.
- [43] R. Gholami and N. Fakhari, "Support vector machine: Principles, parameters, and applications." *Handbook of Neural Computation*, pp. 515–535, 2017.

-
- [44] S. B. Jørgensen and K. M. Hangos, “Grey box modelling for control: Qualitative models as a unifying framework,” *International Journal of Adaptive Control and Signal Processing*, p. 547–562, 1995.
- [45] I. Ahmad, A. Ayub, M. Kano, and I. I. Cheema, “Gray-box soft sensors in process industry: Current practice, and future prospects in era of big data,” *Processes*, vol. 8, no. 2, 2020. [Online]. Available: <https://www.mdpi.com/2227-9717/8/2/243>
- [46] A. Vabalas, E. Gowen, E. Poliakoff, and A. J. Casson, “Machine learning algorithm validation with a limited sample size,” *PLOS ONE*, vol. 14, no. 11, pp. 1–20, November 2019.
- [47] Z. Shen, *K-Fold Cross Validation*, 2020, available at https://zitaoshen.rbind.io/project/machine_learning/machine-learning-101-cross-validation/.
- [48] H. Byeon, “Developing a predictive model for depressive disorders using stacking ensemble and naive bayesian nomogram: Using samples representing south korea,” *Frontiers in Psychiatry*, vol. 12, p. 773290, January 2022.
- [49] J. Mendes, R. Maia, R. Araújo, and F. A. A. Souza, “Self-evolving fuzzy controller composed of univariate fuzzy control rules,” *Applied Sciences*, vol. 10, no. 17, p. 5836, August 2020.
- [50] L. Shi, J. A. Westerhuis, J. Rosén, R. Landberg, and C. Brunius, “Variable selection and validation in multivariate modelling,” *Bioinformatics*, vol. 35, pp. 972–980, 2019.
- [51] L. Yu, K. K. Lai, S. Wang, and W. Huang, “A bias-variance-complexity trade-off framework for complex system modeling,” in *Computational Science and Its Applications - ICCSA 2006*. Springer Berlin Heidelberg, 2006, pp. 518–527.
- [52] S. Fortmann-Roe, *Understanding the Bias-Variance Tradeoff*, June 2012, available at <https://stats.stackexchange.com/questions/553286/apart-from-the-bias-variance-decomposition-is-there-a-bias-variance-proof>.
- [53] Schreiber-Gregory and D. N., “Ridge regression and multicollinearity: An in-depth review,” *Model Assisted Statistics and Applications*, vol. 13, no. 4, pp. 359–365, 2018.
- [54] J. Ranstam and J. A. Cook, “LASSO regression,” *British Journal of Surgery*, vol. 105, no. 10, pp. 1348–1348, 08 2018.
- [55] H. Zou and T. Hastie, “Regularization and variable selection via the elastic net,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 67, no. 2, pp. 301–320, 2005.
-

Bibliografia

- [56] F. Zhang and L. J. O'Donnell, "Chapter 7 - support vector regression," in *Machine Learning*, A. Mechelli and S. Vieira, Eds. Academic Press, 2020, pp. 123–140.
- [57] J. Mendes, F. Souza, R. Araújo, and S. Rastegar, "Neo-fuzzy neuron learning using back-fitting algorithm," *Neural Computing and Applications*, vol. 31, no. 8, pp. 3609–3618, August 2019.
- [58] J. Mendes, F. A. A. Souza, R. Maia, and R. Araújo, "Iterative learning of multiple univariate zero-order t-s fuzzy systems," in *Proc. of The IEEE 45th Annual Conference of the Industrial Electronics Society (IECON 2019)*. Lisbon, Portugal: IEEE, October 14-17 2019, pp. 3657–3662.
- [59] J. H. Friedman, "Multivariate adaptive regression splines," *The Annals of Statistics*, vol. 19, no. 1, pp. 1–67, 1991. [Online]. Available: <http://www.jstor.org/stable/2241837>
- [60] D. Harrison and D. L. Rubinfeld, "Hedonic housing prices and the demand for clean air," *Journal of Environmental Economics and Management*, vol. 5, no. 1, pp. 81–102, 1978. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/0095069678900062>
- [61] M. Gonçalves, P. Sousa, J. Mendes, M. Danishvar, and A. Mousavi, "Real-time event-driven learning in highly volatile systems: A case for embedded machine learning for scada systems," *IEEE Access*, vol. 10, pp. 50 794–50 806, 2022.
- [62] M. Danishvar, S. Danishvar, F. Souza, P. Sousa, and A. Mousavi, "Coarse return prediction in a cement industry's closed grinding circuit system through a fully connected deep neural network (fcdnn) model," *Applied Sciences*, vol. 11, no. 4, 2021. [Online]. Available: <https://www.mdpi.com/2076-3417/11/4/1361>
- [63] T. J. McAvoy, E. H. Hsu, and S. S. Lowenthal, "Dynamics of ph in controlled stirred tank reactor," *Industrial Engineering Chemistry Process Design and Development*, vol. 11, pp. 71–78, 1972.



Anexo A - Gráficos Modelos de Regressão

Conteúdo

A.1	Box-Jenkins	60
A.2	Friedman	62
A.3	Boston Housing	64
A.4	Debutanizer Column	66
A.5	Cement Industry	69
A.6	pH	71

A. Anexo A - Gráficos Modelos de Regressão

No Appendix A são apresentados as figuras dos resultados, em condições de treino e teste, dos modelos de regressão que não foram apresentados no Capítulo 4, para os diferentes conjuntos de dados.

A.1 Box-Jenkins

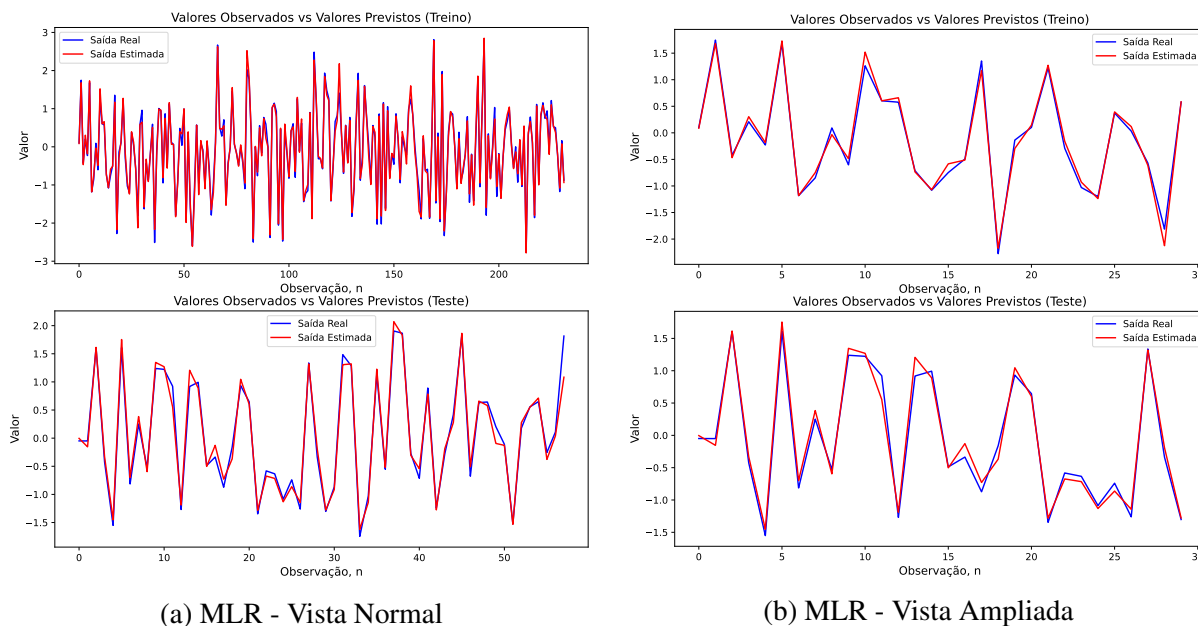


Figura A.1: Resultados de treino e teste do MLR (Box-Jenkins).

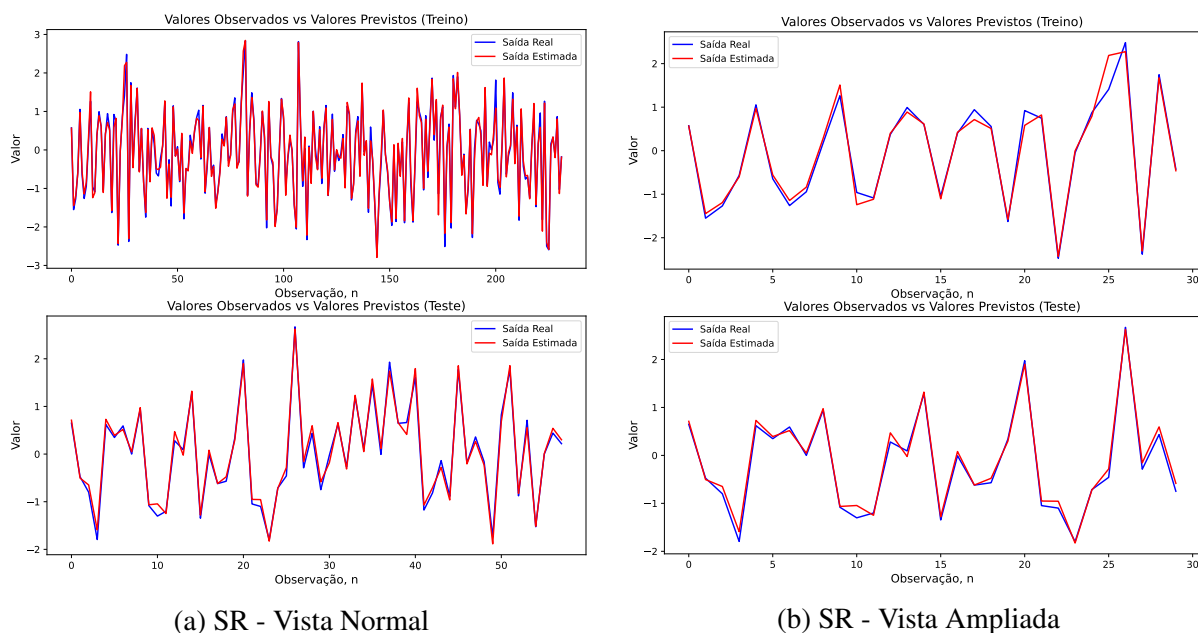
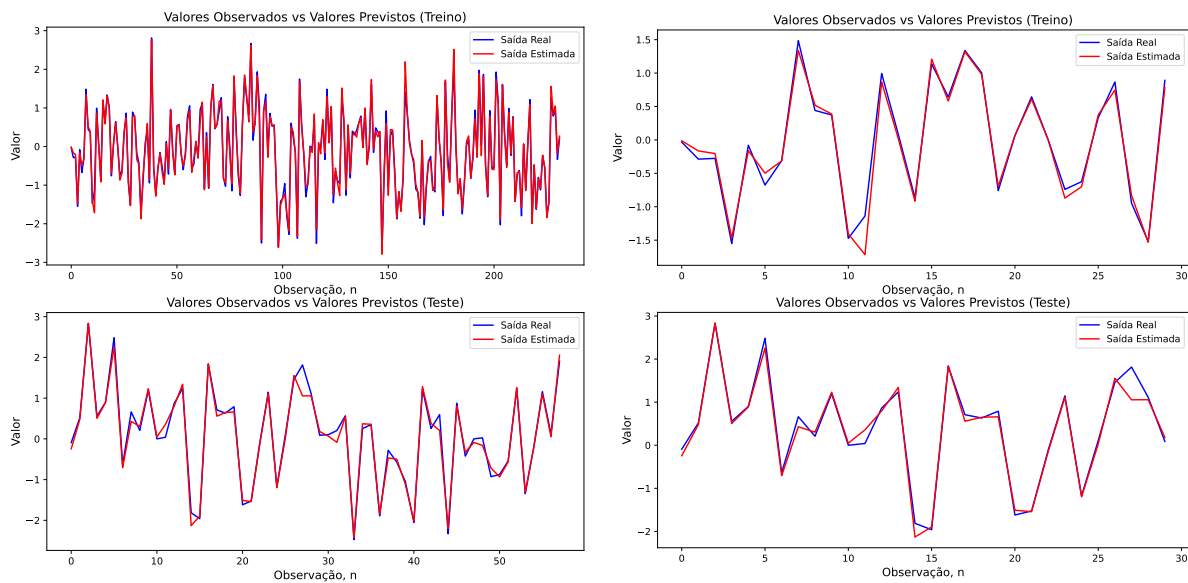


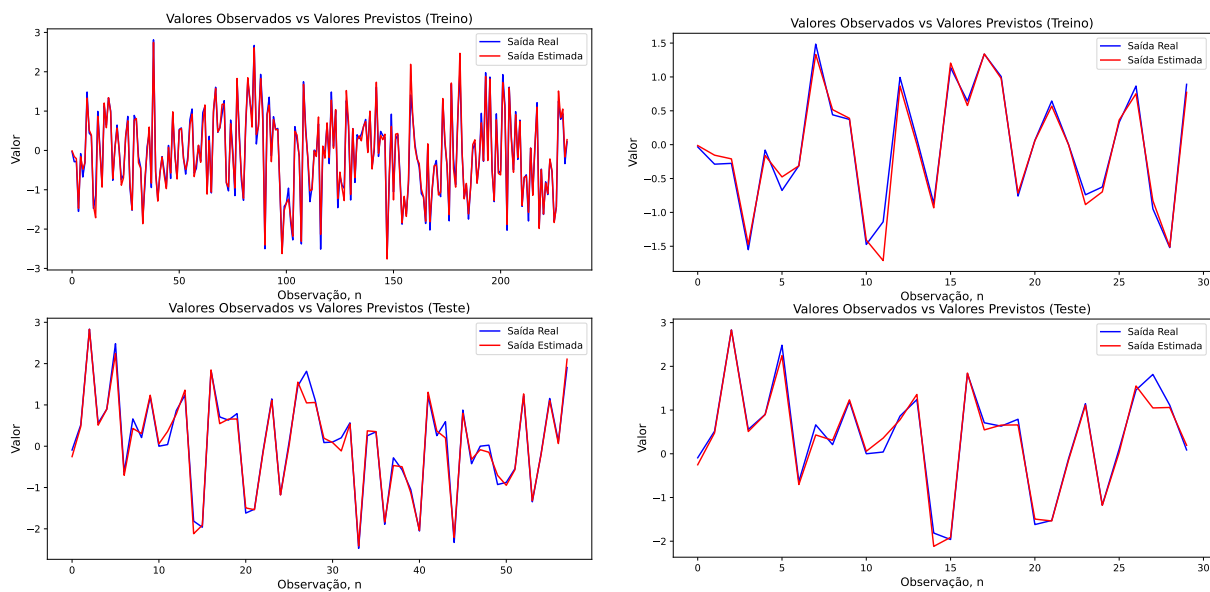
Figura A.2: Resultados de treino e teste do SR (Box-Jenkins).



(a) LASSO - Vista Normal

(b) LASSO - Vista Ampliada

Figura A.3: Resultados de treino e teste do LASSO (Box-Jenkins).



(a) EN - Vista Normal

(b) EN - Vista Ampliada

Figura A.4: Resultados de treino e teste do EN (Box-Jenkins).

A. Anexo A - Gráficos Modelos de Regressão

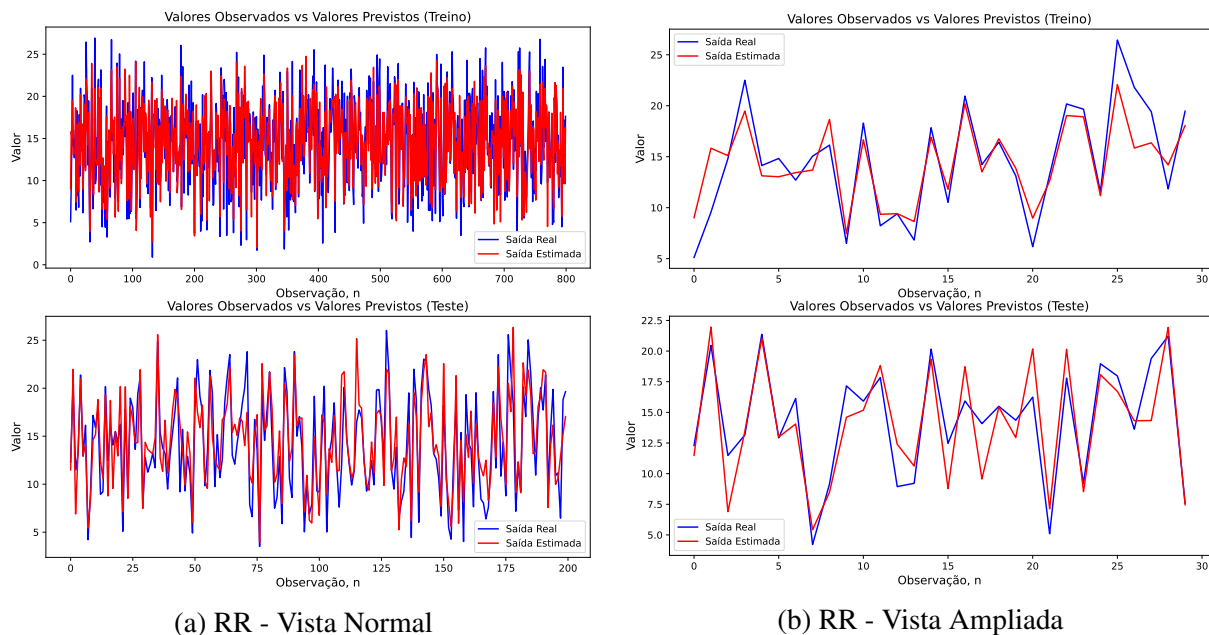


Figura A.5: Resultados de treino e teste do RR (Friedman).

A.2 Friedman

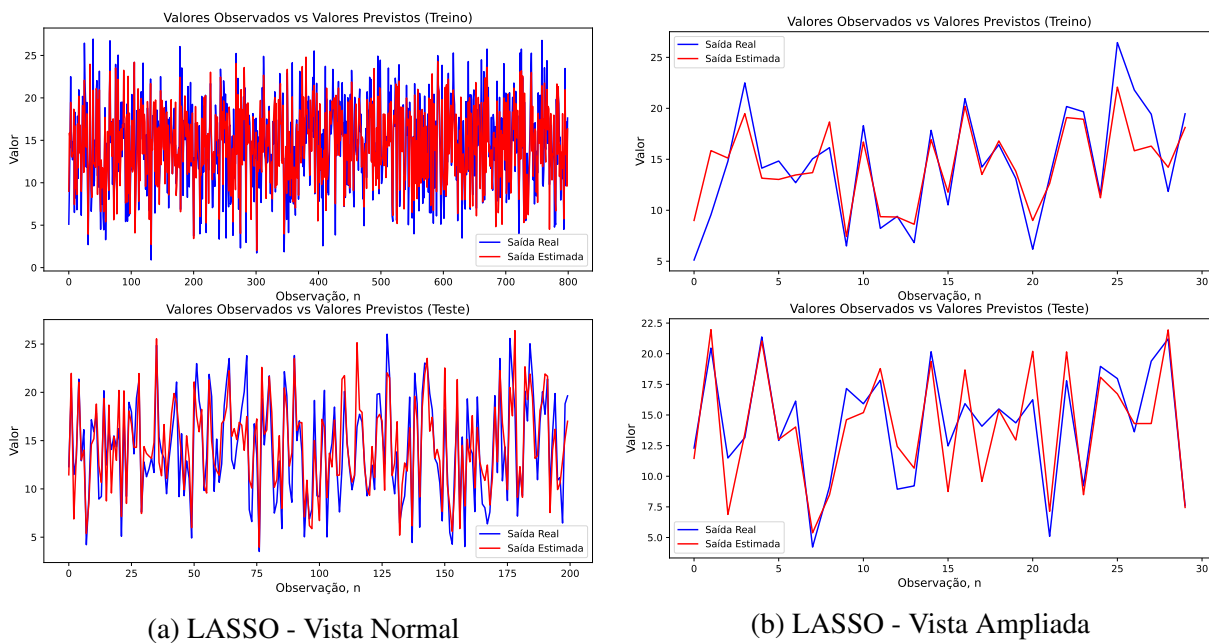


Figura A.6: Resultados de treino e teste do LASSO (Friedman).

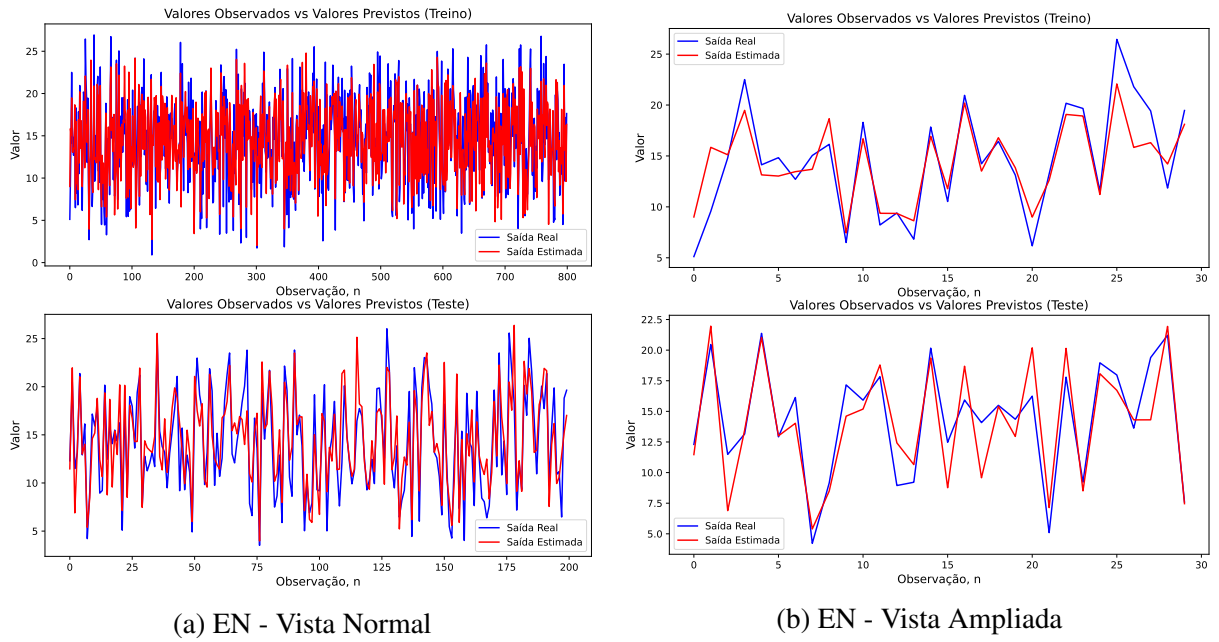


Figura A.7: Resultados de treino e teste do EN (Friedman).

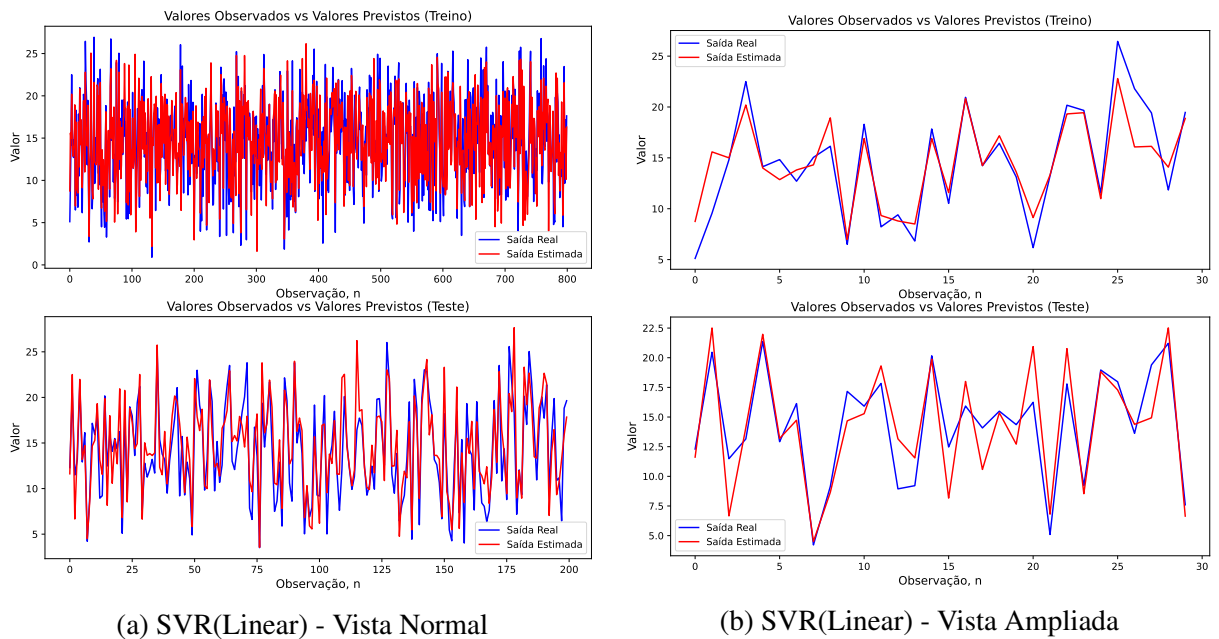


Figura A.8: Resultados de treino e teste do SVR(Linear) (Friedman).

A. Anexo A - Gráficos Modelos de Regressão

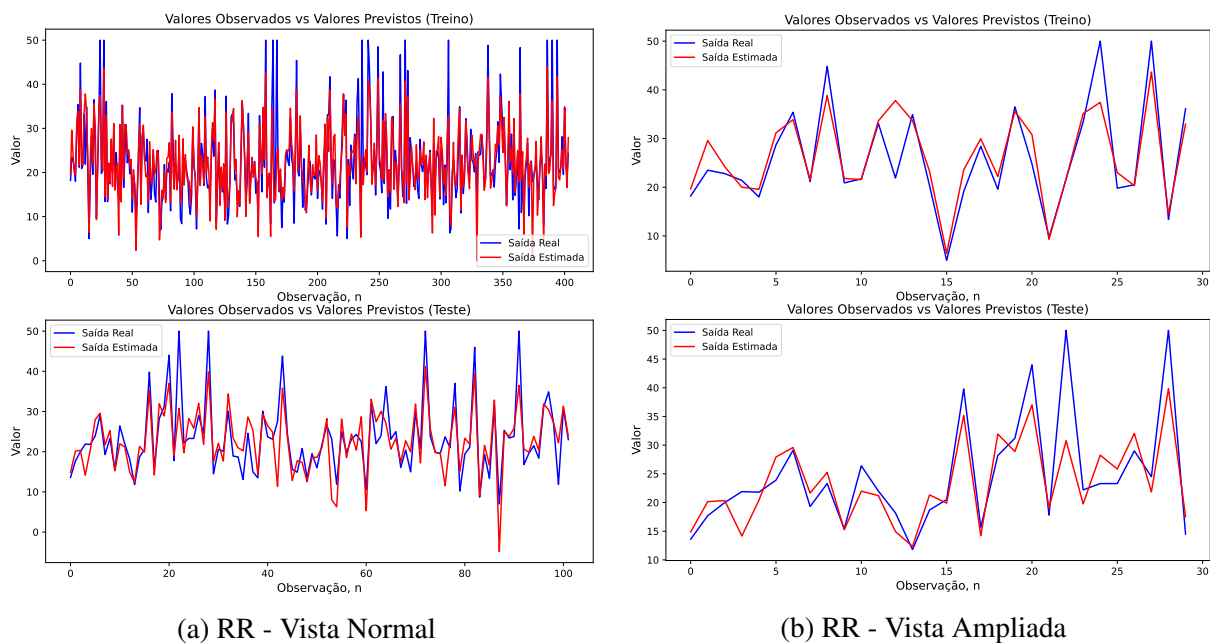


Figura A.9: Resultados de treino e teste do RR (Boston Housing).

A.3 Boston Housing

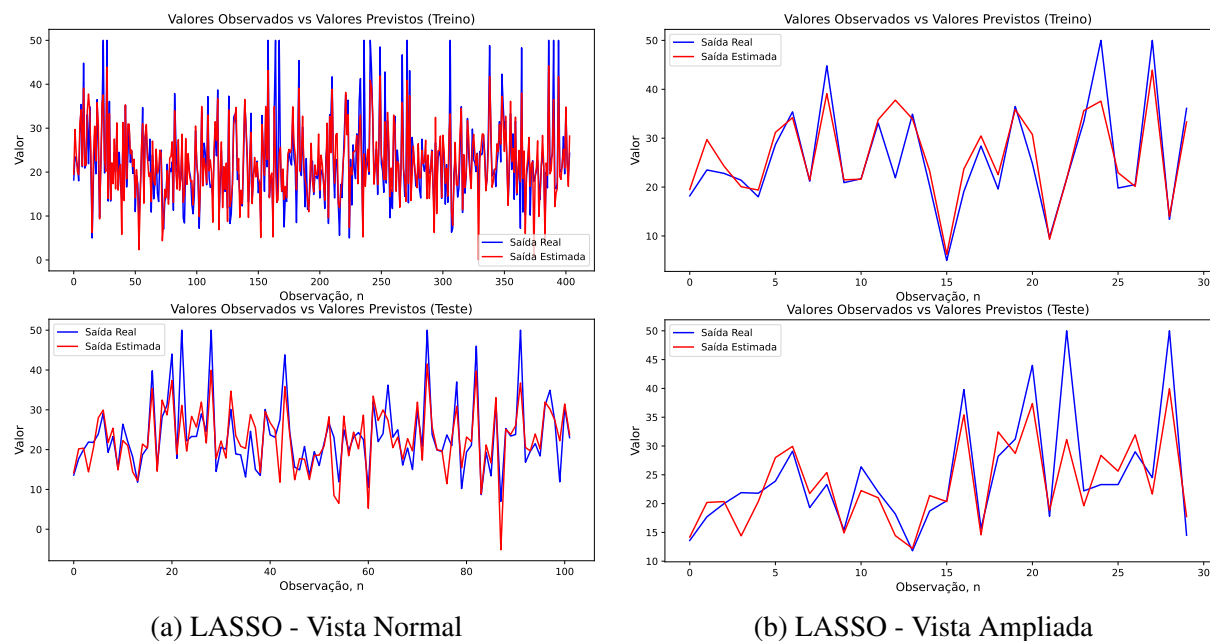


Figura A.10: Resultados de treino e teste do LASSO (Boston Housing).

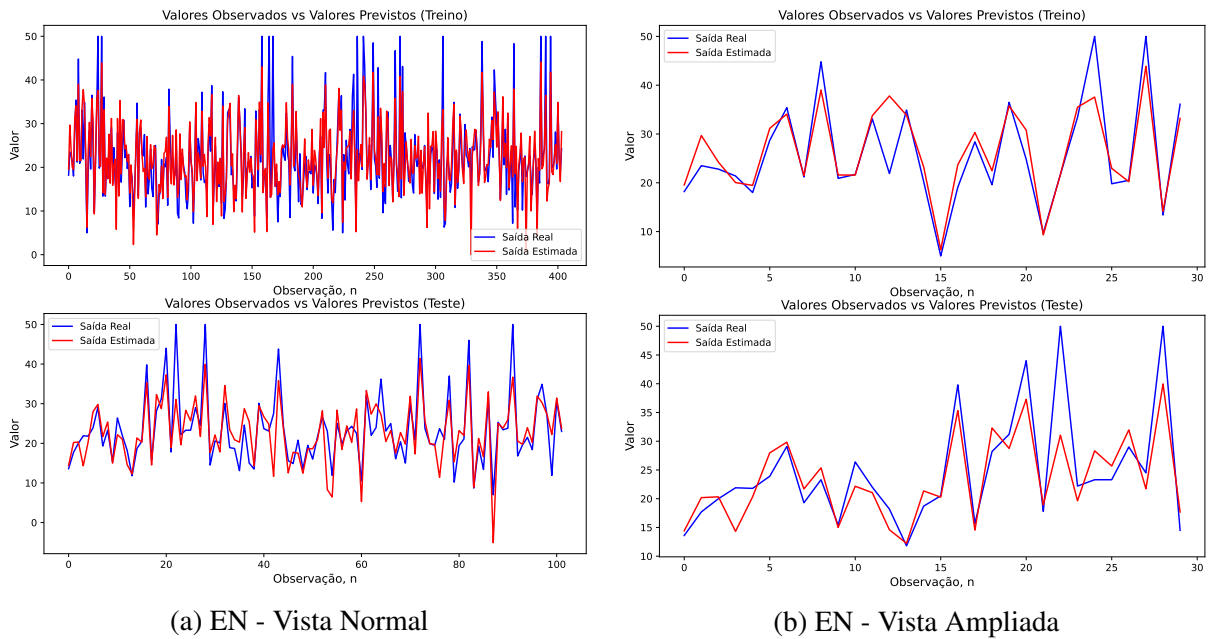


Figura A.11: Resultados de treino e teste do EN (Boston Housing).

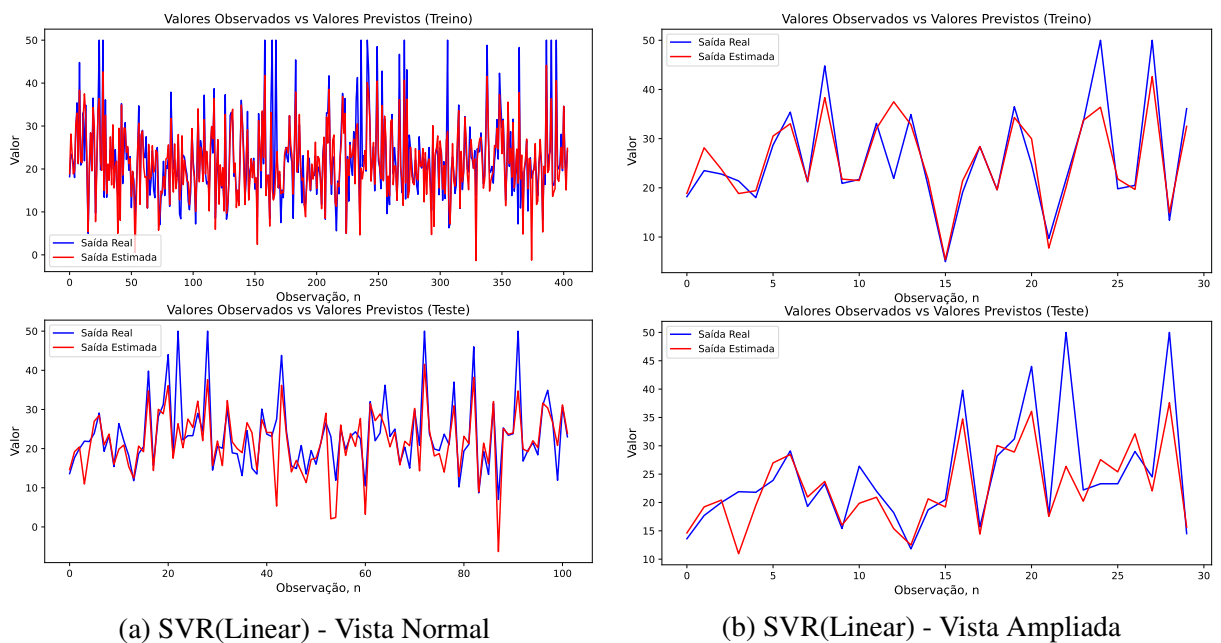


Figura A.12: Resultados de treino e teste do SVR(Linear) (Boston Housing).

A. Anexo A - Gráficos Modelos de Regressão

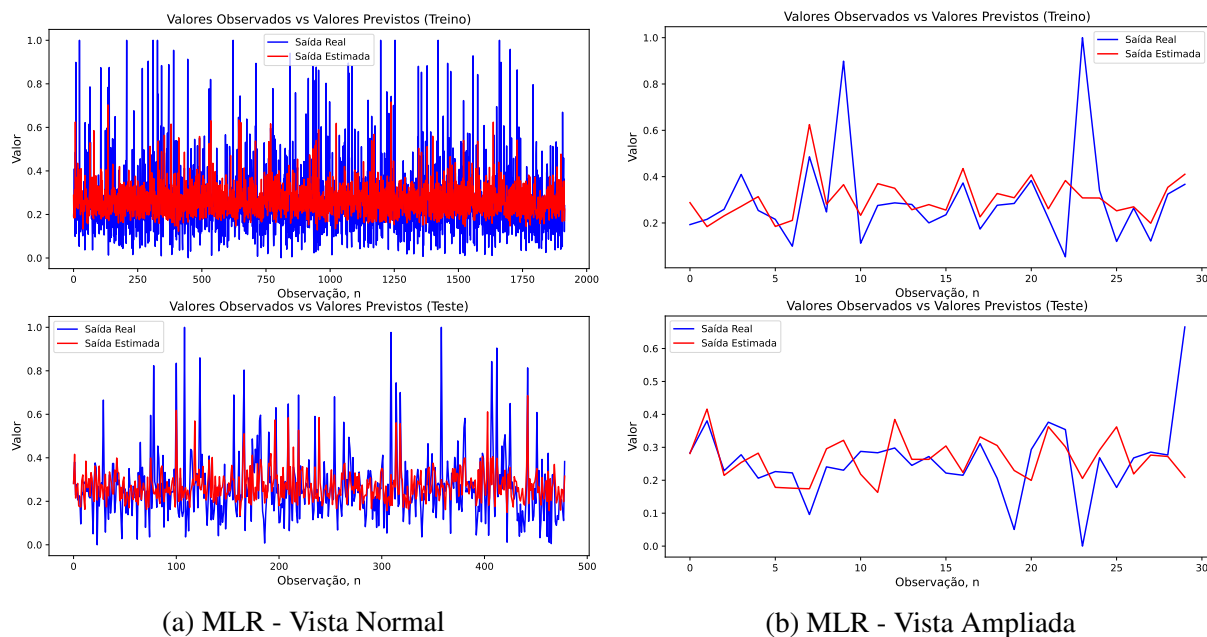


Figura A.13: Resultados de treino e teste do MLR (Debutanizer Column).

A.4 Debutanizer Column

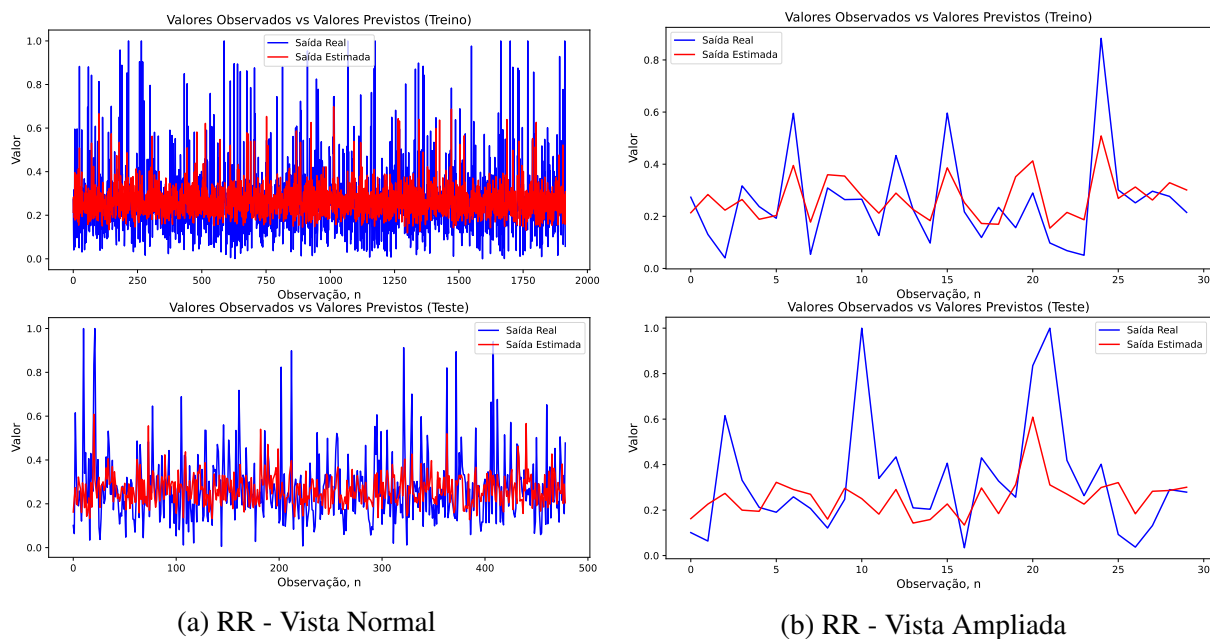


Figura A.15: Resultados de treino e teste do LASSO (Debutanizer Column).

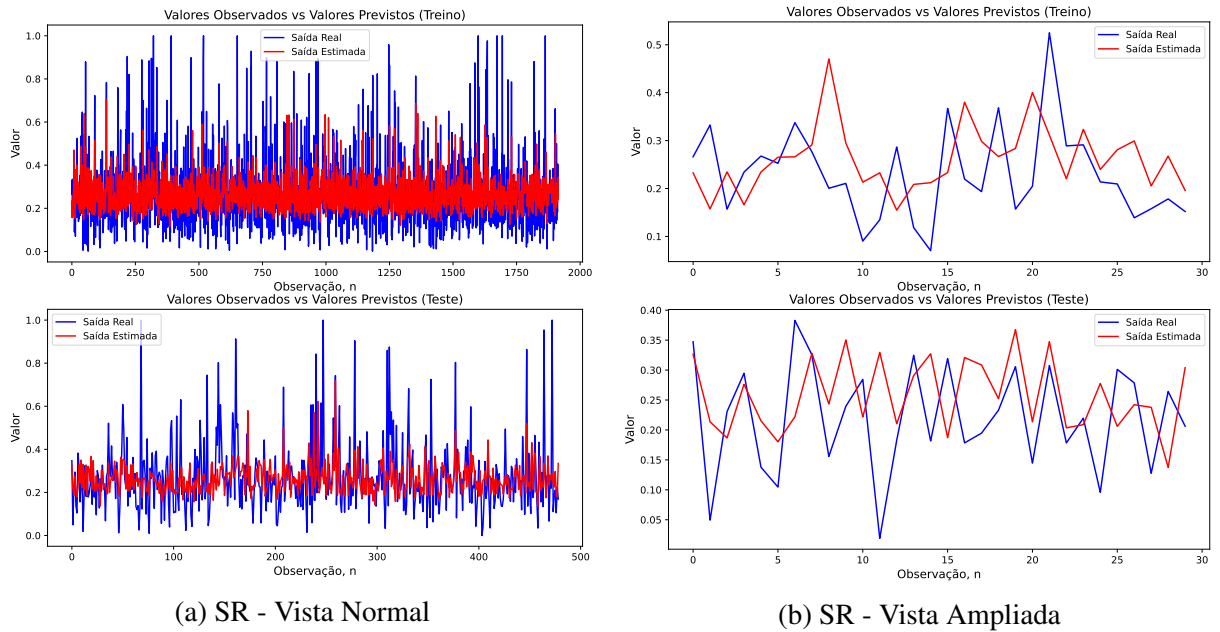


Figura A.14: Resultados de treino e teste do SR (Debutanizer Column).

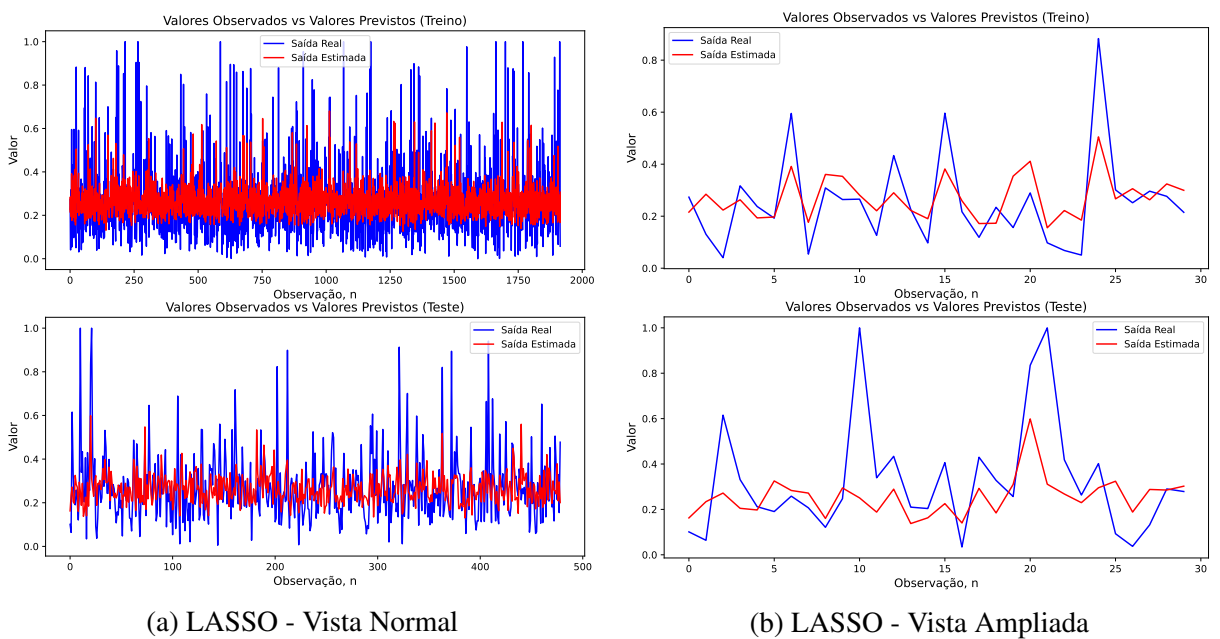
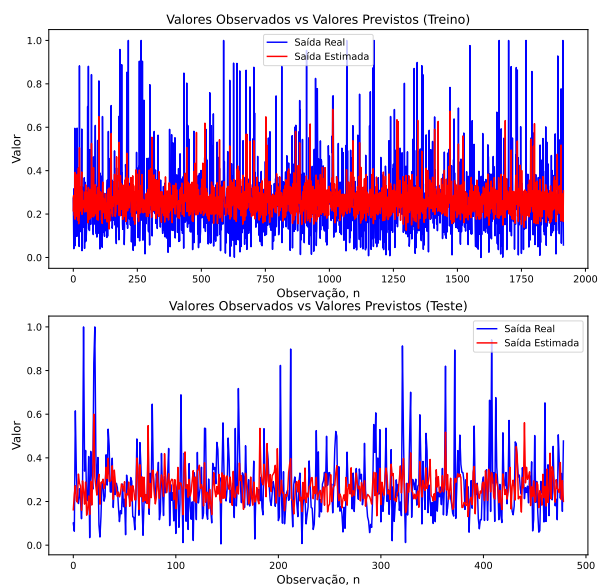
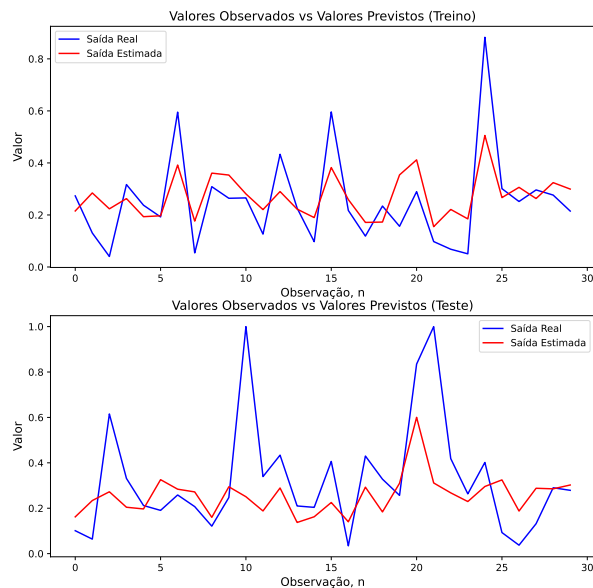


Figura A.16: Resultados de treino e teste do LASSO (Debutanizer Column).

A. Anexo A - Gráficos Modelos de Regressão

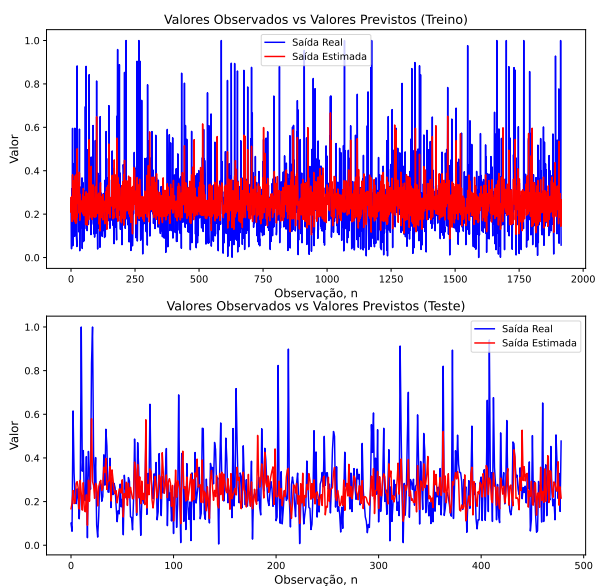


(a) EN - Vista Normal

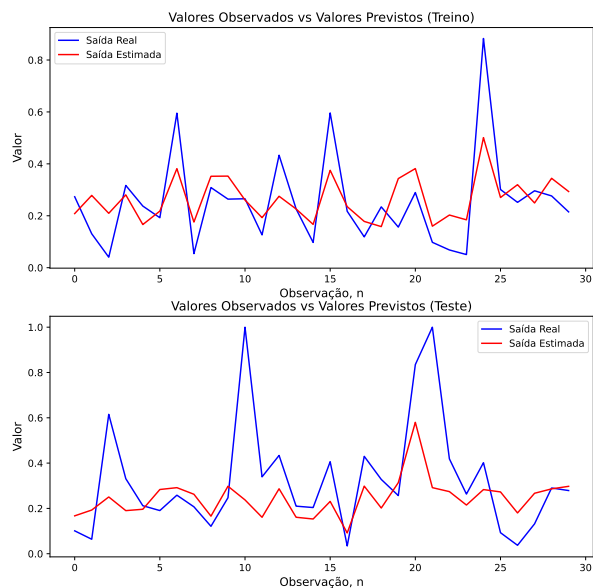


(b) EN - Vista Ampliada

Figura A.17: Resultados de treino e teste do EN (Debutanizer Column).



(a) SVR(Linear) - Vista Normal



(b) SVR(Linear) - Vista Ampliada

Figura A.18: Resultados de treino e teste do SVR(Linear) (Debutanizer Column).

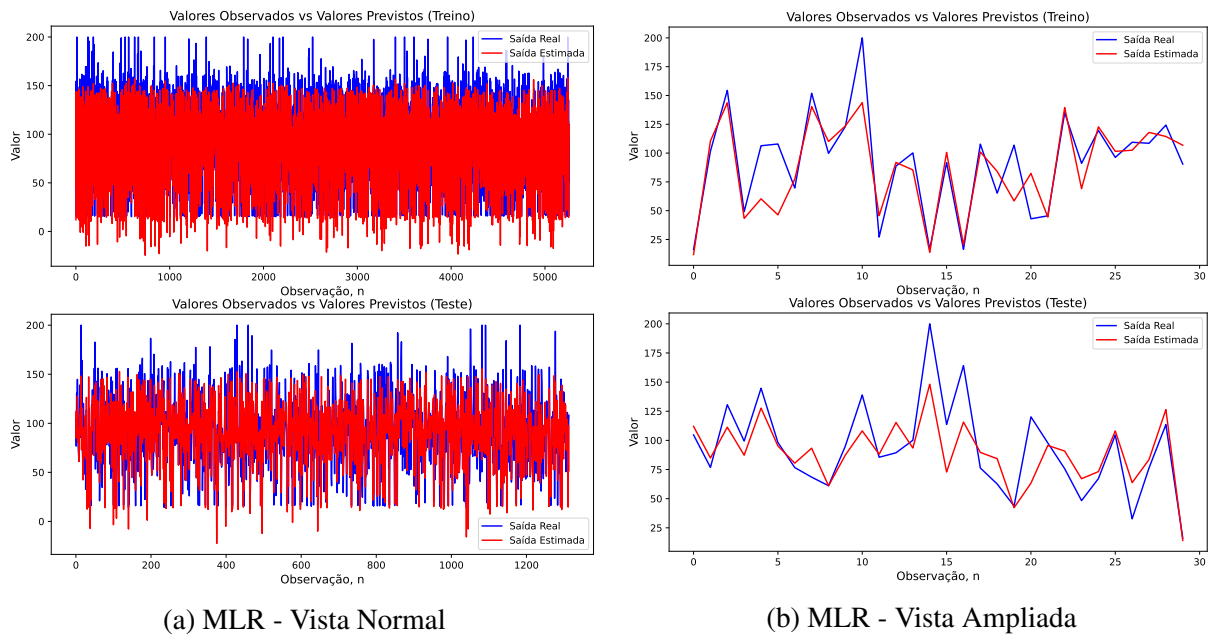


Figura A.19: Resultados de treino e teste do MLR (Cement Industry).

A.5 Cement Industry

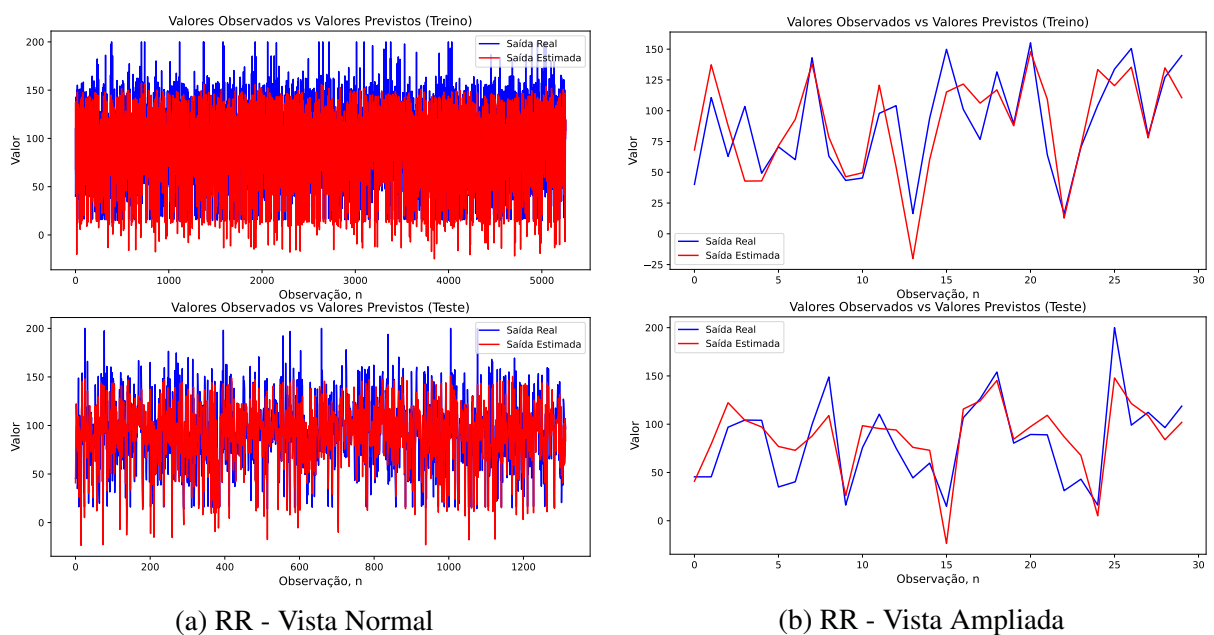
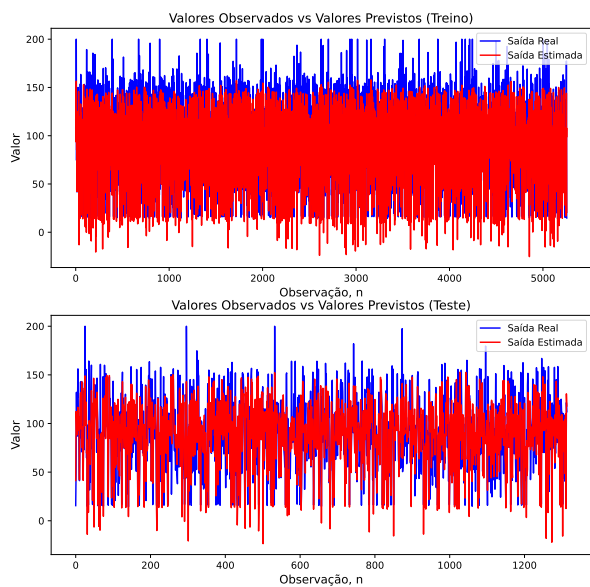
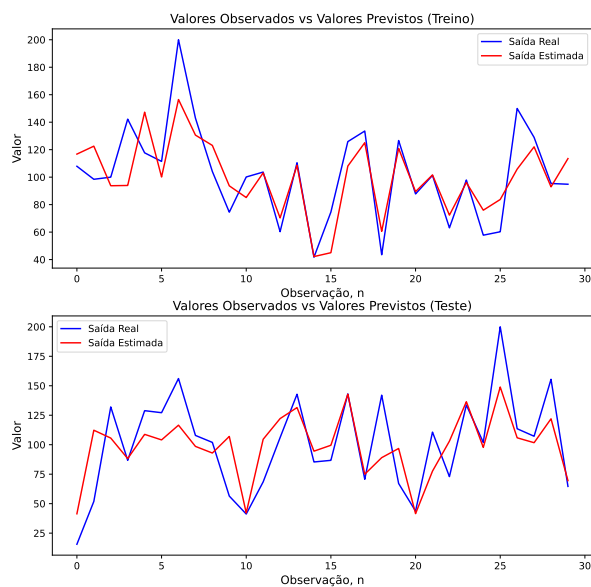


Figura A.21: Resultados de treino e teste do LASSO (Cement Industry).

A. Anexo A - Gráficos Modelos de Regressão

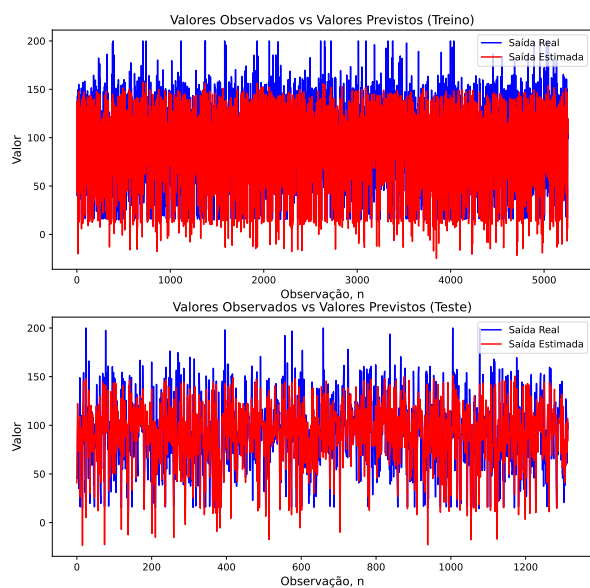


(a) SR - Vista Normal

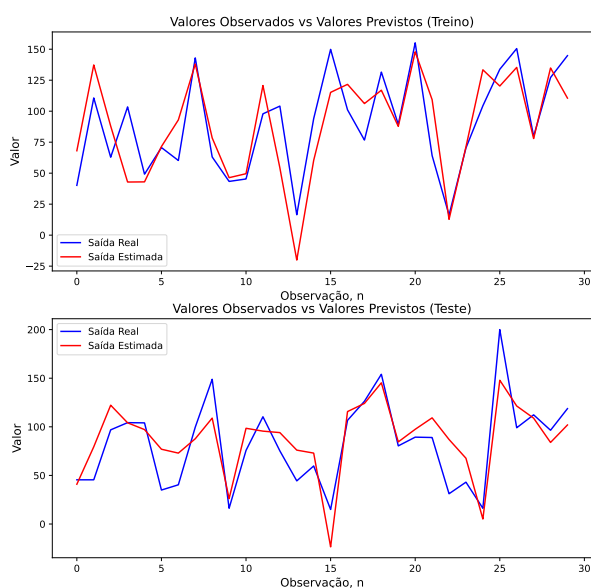


(b) SR - Vista Ampliada

Figura A.20: Resultados de treino e teste do SR (Cement Industry).

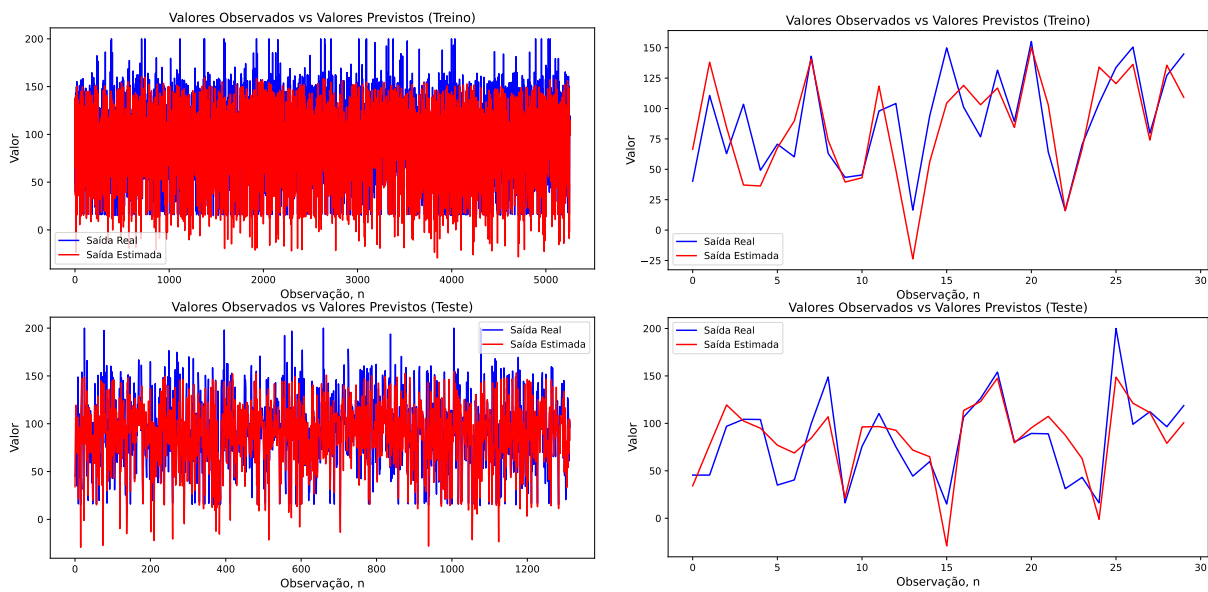


(a) EN - Vista Normal



(b) EN - Vista Ampliada

Figura A.22: Resultados de treino e teste do EN (Cement Industry).

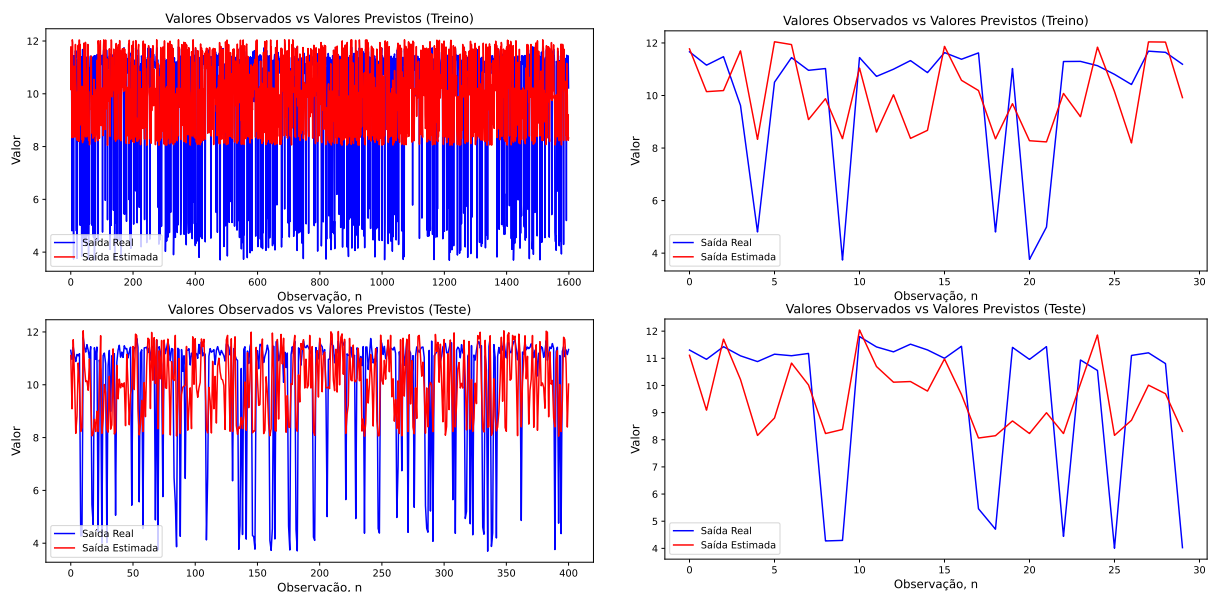


(a) SVR(Linear) - Vista Normal

(b) SVR(Linear) - Vista Ampliada

Figura A.23: Resultados de treino e teste do SVR(Linear) (Cement Industry).

A.6 pH



(a) MLR - Vista Normal

(b) MLR - Vista Ampliada

Figura A.24: Resultados de treino e teste do MLR (pH).

A. Anexo A - Gráficos Modelos de Regressão

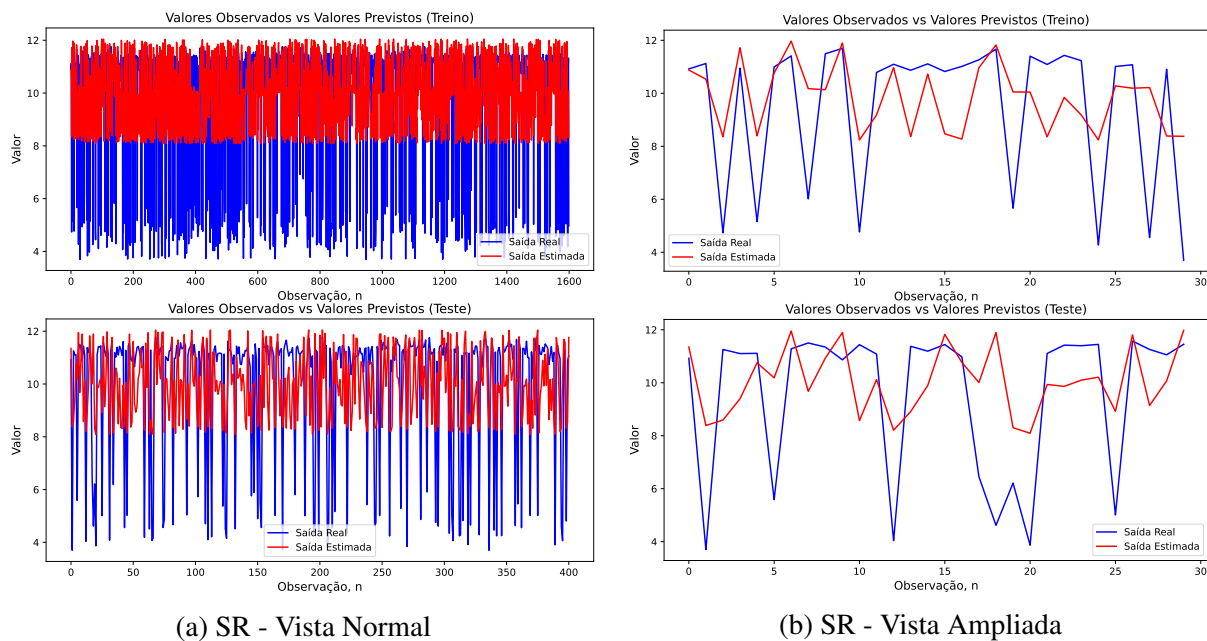


Figura A.25: Resultados de treino e teste do SR (pH).

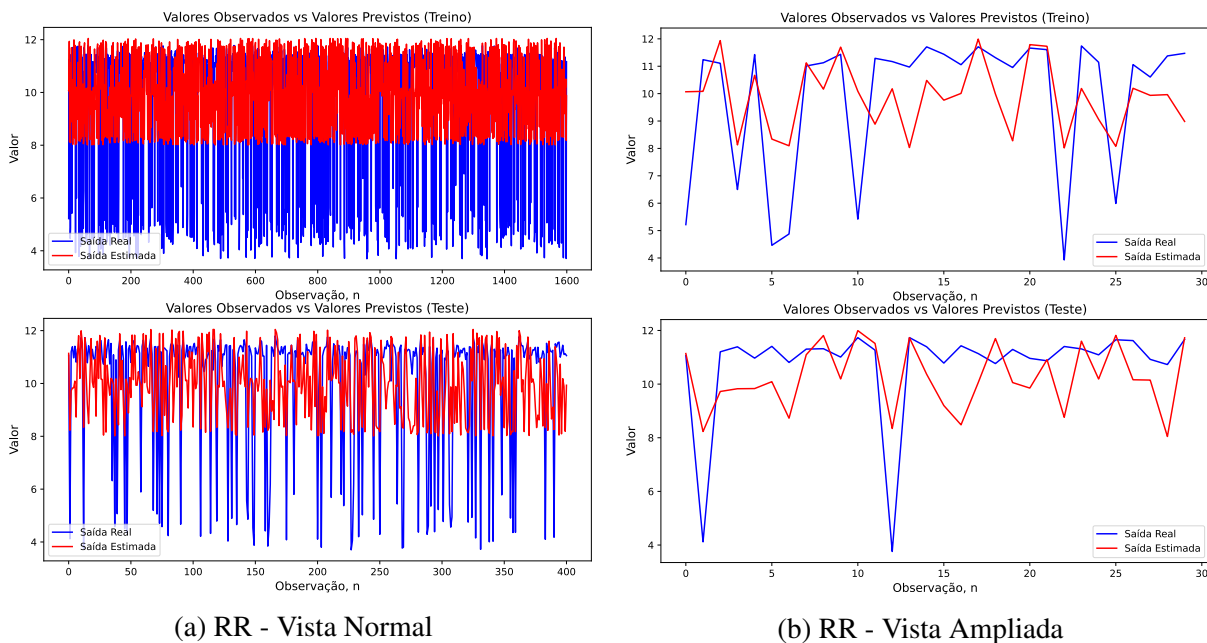


Figura A.26: Resultados de treino e teste do RR (pH).

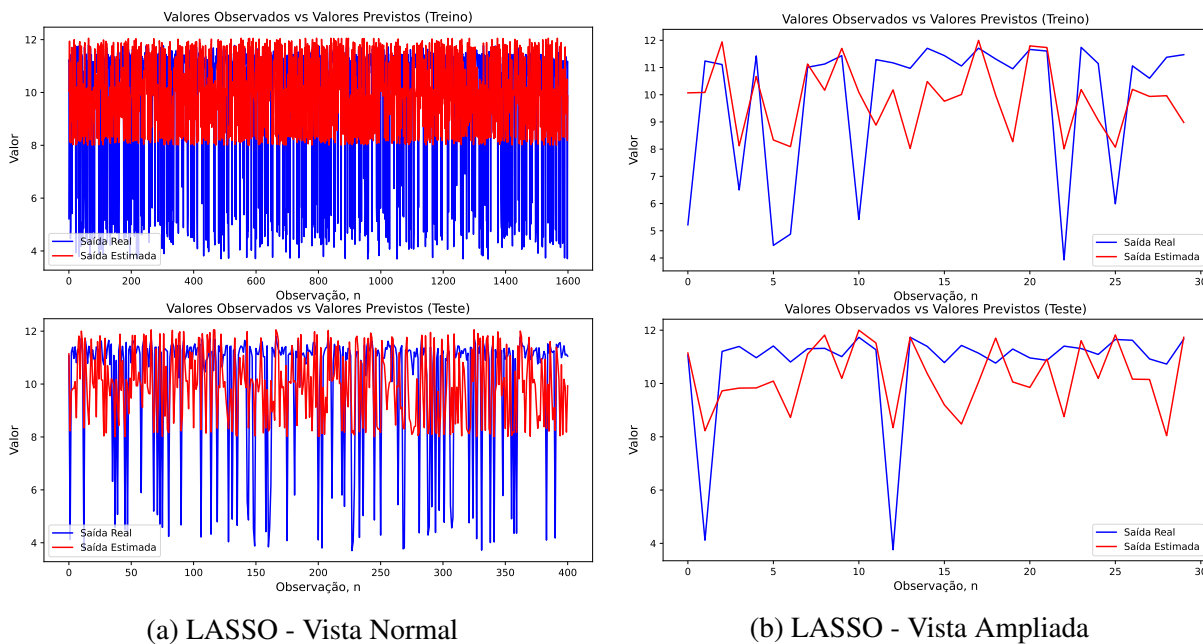


Figura A.27: Resultados de treino e teste do EN (pH).

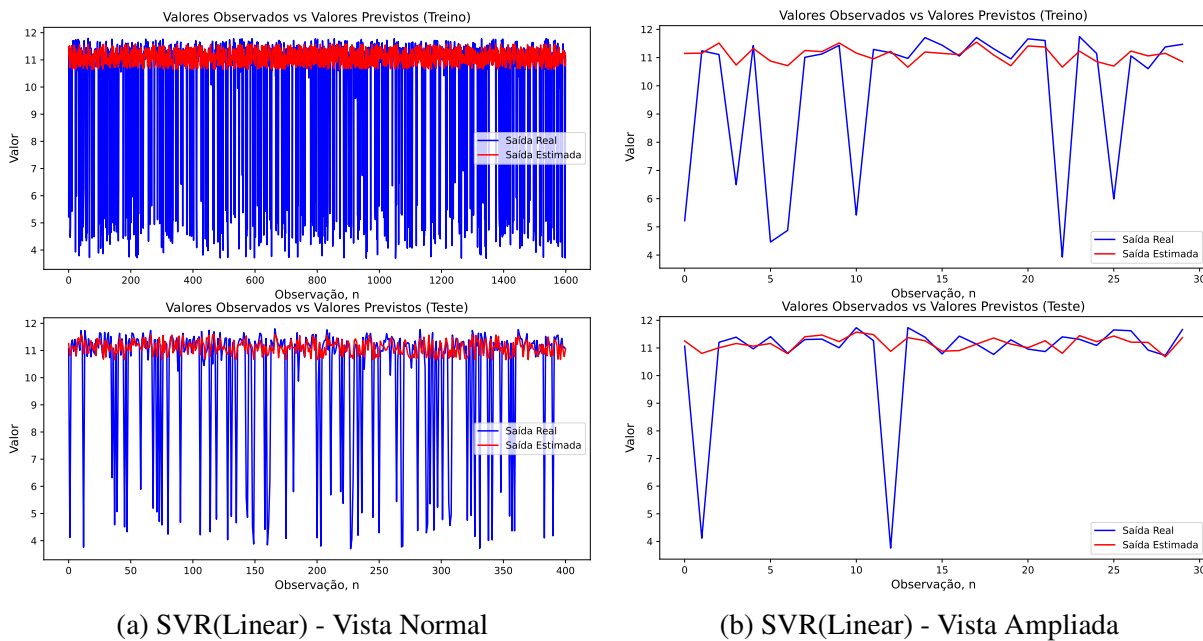


Figura A.28: Resultados de treino e teste do SVR(Linear) (ph).

